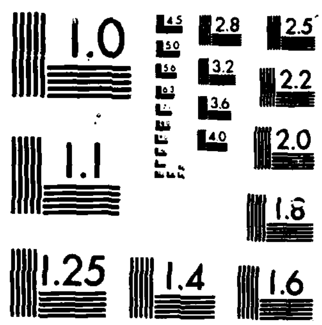
AD-A164 496 MULTICHANNEL COLLISION RESOLUTION ALGORITHM(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA J H CHO DEC 85 1/2

UNCLASSIFIED F/G 17/2 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3

(2)

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

MULTICHANNEL COLLISION RESOLUTION ALGORITHM

by

Joo Hyung Cho

December 1985

Thesis Advisor: J. F. Chang

Approved for public release; distribution is unlimited.

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL (If applicable) 62	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5100	7b. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5100
8a NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
MULTICHANNEL COLLISION RESOLUTION ALGORITHM

12 PERSONAL AUTHOR(S)
Cho, Joo Hyung

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM ______ TO ______	1985 December	99

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Multichannel System; Static m-ary Splitting; Dynamic m-ary Splitting; Binary Splitting with Energy Detector; Collision Resolution Algorithm

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

A multichannel random accessed communication system was considered. Every time slotted channel is shared by a large number of transmitters. To solve a collision which occurs at a certain time slot, several collision resolution protocols were tested, analyzed, and simulated. The quantities to be investigated in this thesis include $\bar{L}_n$ (the average number of slots required to resolve a collision involving n users), linear bounding of $\bar{L}_n$ and the maximal achievable throughput.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL Hung-Mou Lee	22b TELEPHONE (Include Area Code) 408 646-2030 22c OFFICE SYMBOL 62Lh

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

Multichannel Collision Resolution Algorithm

by

Cho, Joo Hyung
Major, Republic of Korea Air Force
B.S., R.O.K. Air Force Academy, 1975

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1985

Author: Cho, Joo Hyung

Approved by: Jin-Fu Chang, Thesis Advisor

Hung-Mou Lee, Second Reader

Harriett B. Rigas, Chairman,
Department of Electrical and Computer Engineering

John N. Dyer,
Dean of Science and Engineering

ABSTRACT

A multichannel random accessed communication system was considered. Every time slotted channel is shared by a large number of transmitters. To solve a collision which occurs at a certain time slot, several collision resolution protocols were tested, analyzed, and simulated. The quantities to be investigated in this thesis include $\bar{L}_n$ (the average number of slots required to resolve a collision involving n users), linear bounding of $\bar{L}_n$ and the maximal achievable throughput.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to my thesis advisor, Prof. Jin-Fu Chang for the encouragement, guidance and understanding which contributed to the completion of this thesis. Also, I wish to extend special thanks to my wife, daughter and to all others who have provided invaluable support during my study at NPS. Finally, I would like to thank the Korean Air Force Headquarters for giving me an opportunity to study here.

I. INTRODUCTION

The purpose of this thesis is to propose and analyze a few collision resolution protocols suitable for multichannel environment. To have some familiarity with collision resolution algorithms, the computer communication networks and the progression of the algorithms should be explained.

A. BACKGROUND

1. Computer Networks

In the past several decades, we have witnessed the birth and unprecedented growth of the computer industry. In this spectacular progression of the computer industry, computers have become smaller, cheaper and more numerous.

The increased use of computers in the communications field has greatly influenced the design and architecture of computer systems and networks. In the past, computing was generally performed at a centralized facility, using a mainframe computer. Present day systems increasingly rely on distributed processing using smaller remotely located computers connected by a network. These systems are called "computer networks". A computer network can be defined as an interconnected group of independent computer systems which communicate with one another. These computer systems share resources such as programs, data, hardware, or software. Obviously, the goals of computer networks are resource sharing, load balancing, high reliability, and the reduction of communication costs. Finally, it can provide a powerful communication medium among widely separated people.

Figure 1.1 shows the conceptual model of a computer network. Two basic components of the communication subnet are switching elements and transmission lines. The communication subnet is responsible for transmitting messages reliably and rapidly between all participating network sites.

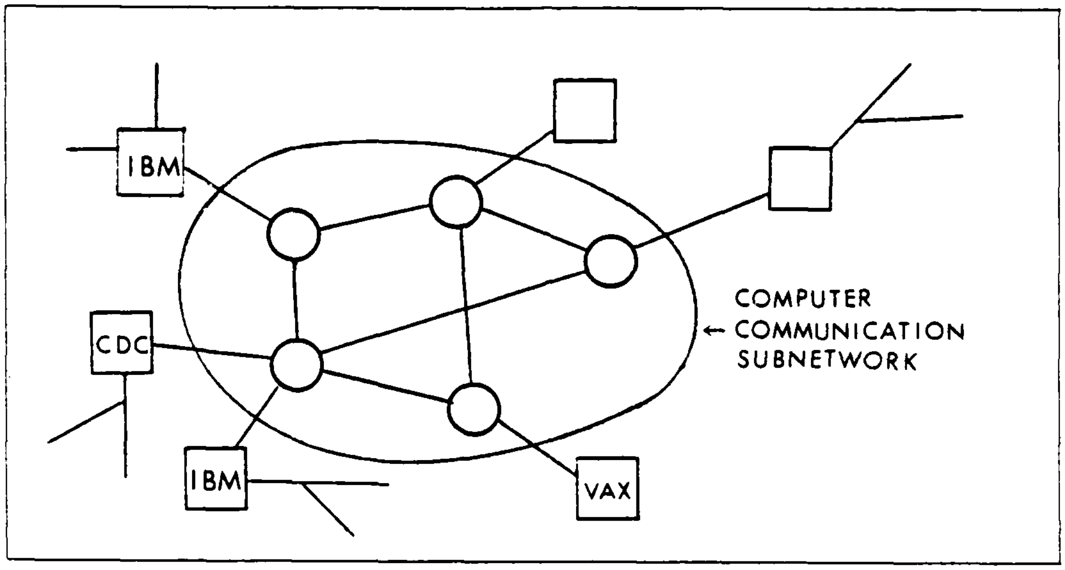


Figure 1.1 Computer Network

2. Computer Communication Networks

A computer network consists of communication devices, the host processors, the transmission lines and a set of rules. These rules can be implemented in either hardware or software to insure the orderly flow of traffic in the network. As computer networks are expanded, communication between computers has become more complicated and more important. For data communication in a computer communication network, several switching techniques can be used. These are circuit switching, message switching, and packet switching.

Circuit switching is very similar to telephone conversation and once a route has been set up, it will be used for the entire conversation. Message switching is another switching strategy. A transaction between two users in the network can be visualized as a conversation and each message is transmitted independently in the manner of store-and-forward. In packet switching, each message breaks into several fixed-length data units called packets. Main advantages of packet switching include reduced delay and improved throughput. It is particularly suitable in handling interactive traffics such as data generated by computers. Therefore, computer communication is usually done on the basis of packet switching.

3. Problems Involved in Computer Communication Network

There are several problems that should be considered in designing a computer communication network. These include network topology, routing algorithm, congestion control, capacity allocation, concentrator siting and network access techniques.

In this thesis, network access techniques will be studied. It is desirable, if interference can be suppressed, to transmit and receive a number of signals over a single communication path. By sharing the same communications path with other signals adopting such a "multiple access" technique may effectively reduce the cost per bit of data transmitted.

4. Multiple Access Techniques

One classical solution to the multiple access problem is multiplexing, e.g., time division multiple access (TDMA) or frequency division multiple access (FDMA). In FDMA, all users share the frequency spectrum of a communications path and transmit simultaneously. Note that

each user is allocated a unique frequency band. If the full bandwidth W of the communications path is divided into N users (or N channels), then each user can transmit only at speeds less than the frequency slot of W/N. The limitation is due to the need for small guard bands between adjacent channels to prevent any sideband signal from overlapping. In TDMA, all users occupy the same RF bandwidth, but transmit sequentially in time.

The next solution of the multiple access problem is random access technique, which permits any user to seize the entire communications resource when he has information to transmit. Random multiple access is one of the most efficient tools for channel sharing by a number of users. Random access technique is expedient when there are a large number of stations with bursty input traffic.

5. Random Access Techniques

In 1970, Abramson [Ref. 14] devised a pure ALOHA system. In this system, a single channel is shared by all users and a user transmits data as soon as he is ready to do so. It is assumed that the data is transmitted in packet form and all transmitted packets have identical length. Each packet contains the information to be transmitted, along with the source and destination addresses and any other overhead information (error correcting and error detecting bits). It is supposed that the transmitter only sends a packet of some fixed duration, say T seconds. During this T-second transmission, if only one transmitter sends a packet, then the transmission is successful. Since there is no synchronization among users, transmission times of two or more users may overlap. This situation is referred to as a collision. However, due to the perfect feedback property of packet broadcasting, the sender of a packet can always find out whether or not his packet was successfully transmitted

by just listening to the channel. A collision results in complete loss of the information carried by the collided packets, thus retransmission of such packets are necessary. To avoid a repetition of the same collision, pure ALOHA specifies that, after a collision, each transmitter involved randomly selects a waiting time before it again retransmits its packet. Throughput of the system can be measured as the average number of successful transmissions during a message transmission time (usually normalized to 1 for convenience), and is given by:

$$S = G\,P_o$$

where S is the throughput, G is the channel load measured as the number of attempted transmissions per unit of normalized time, and P_o is the probability of no collision. Assuming that the channel load G follows Poisson distribution, P_o for pure ALOHA is given by:

$$P_o = e^{-2G}$$

therefore:

$$S = Ge^{-2G}$$

The maximum throughput one can have is 0.184, corresponding to G=0.5.

In 1972, Roberts [Ref. 15] modified Abramson's algorithm. This algorithm is now known as the slotted-ALOHA random access algorithm which doubles the capacity of the pure ALOHA system. This scheme operates exactly the same as the pure ALOHA does, except that a new packet transmission must begin at the next slot boundary. Time is divided into slots with length equal to the packet transmission time. If a packet is generated in the middle of a slot, then it must wait until the next slot boundary before it is transmitted. A collided packet must be retransmitted at some random

future time slot. The probability of no collision in this case is given by:

$$P_o = e^{-G}$$

and $S = Ge^{-G}$ gives a maximum throughput of 0.368 for G=1.

Small increases in the channel load can drastically reduce its performance because the number of collisions increases exponentially with higher values of G. It should be noted that system performance can be evaluated by three properties: throughput, delay, and stability. ALOHA channels are fundamentally unstable, but there exist a number of simple control procedures which stabilize these channels. Most of these control schemes estimate the number of busy terminals in the system. The "Collision Resolution Approach" opens the possibility of an exact analysis of the stable steady-state behavior of a random-access system.

B. HISTORICAL REVIEW OF COLLISION RESOLUTION ALGORITHMS

Over the years, the research on collision resolution algorithms has always focused on the bursty arrivals of messages and the interference between transmitters, but has generally ignored the noise. It has been always assumed that a message can be successfully communicated in the absence of collision. A Collision Resolution Algorithm (CRA) can be defined as an algorithm which organizes the retransmission of colliding packets in such a way that every packet is eventually transmitted successfully with finite delay. Collision resolution interval (CRI) is referred to as the period beginning with the slot containing the original collision (if any) and ending with the slot in which the original collision is resolved (or ending with the first slot when that slot is collision-free). General assumptions for the CRA are as follows.

- The forward channel to the receiver is a single, errorless, time-slotted, collision-type channel.
- The transmitter can send a data in a packet form whose duration is same as one slot.
- A "collision" between two or more packets is always detected, at least, by the participating receivers.
- A noiseless feedback broadcast concerning the status of a slot is always available to all the participating users. The status of a slot could be any one of the following: (a) that slot was empty or (b) that slot contained one packet (which was thus successfully transmitted) or (c) that slot contained a collision of two or more packets.
- Signal propagation delays can be ignored.

1. Capetanakis Collision Resolution Algorithm

The recent discovery by Capetanakis [Ref. 6] of a collision-resolution algorithm was a surprising development in the evolution of random-access techniques. This tree algorithm shall be referred to as the Capetanakis Collision Resolution Algorithm (CCRA). The CCRA can be stated as follows: after a collision, all transmitters involved flip a binary fair coin, those flipping 0 retransmit in the very next slot, those flipping 1 retransmit in the next slot after the collision (if any) among those flipping 0 has been resolved. No new packets are allowed to transmit until after the initial collision has been resolved [Ref. 8:p. 77].

The following example should both clarify the algorithm and illustrate its main features. Suppose that the initial collision is among three users, as shown in Figure 1.2. For convenience, we refer to these users as A, B and C. After the collision in slot 1, all three of these users flip their coins; we suppose that C flips 0 while A and B flip 1. Thus C sends in slot 2, while A and B transmit in slot 3, and a collision occurs. After a collision in slot 3, we suppose that A and B both flip 1. Thus, slot 4 is empty, so A and B again recognize that they should retransmit in slot 5. After a collision in slot 5, we

suppose that A and B both flip 0. Hence they both retransmit in slot 6. After the collision in slot 6, we suppose that A flips 0 and B flips 1. Thus, A successfully transmits in slot 7 and B successfully transmits in slot 8. All three users in the original collision have now transmitted successfully, but the collision is not yet resolved. The reason for this is that no one can be sure that there was not another user who transmitted in slot 1, then flipped 1 in collision slot 3 and 5 and who thus is now waiting to retransmit in slot 9. It is not until slot 9 proves to be empty that the original collision is finally resolved.

Figure 1.3 shows the tree diagram which illustrates the same situation as Figure 1.2. The number inside the node indicates the feedback information for that time slot (0 = empty slot, 1 = single packet, ≥2 = collision), and the binary numbers on the branches coming from a node indicate the path followed by these users that flipped that binary number after the collision at that node. The number above each node indicates the time slot. At slot 9, all users now simultaneously learn that all the packets in the original collision have been successfully transmitted because a binary rooted tree has two more terminal nodes than it has intermediate nodes excluding the root node. This is the property of CCRA. Additionally, all transmitters must know when the original collision (if any) has been resolved as this determines when new packets may be sent.

In the implementation of CCRA, the feedback information is used only to determine whether the corresponding slot has a collision or is collision-free. Therefore, it is not important whether feedback information is empty or successful (only one packet), as both cases mean that the slot is collision-free.

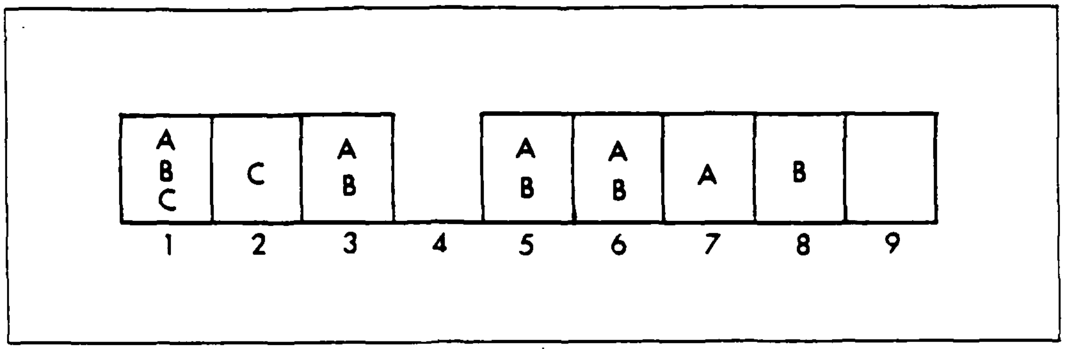


Figure 1.2 Collision Resolution Interval for CCRA

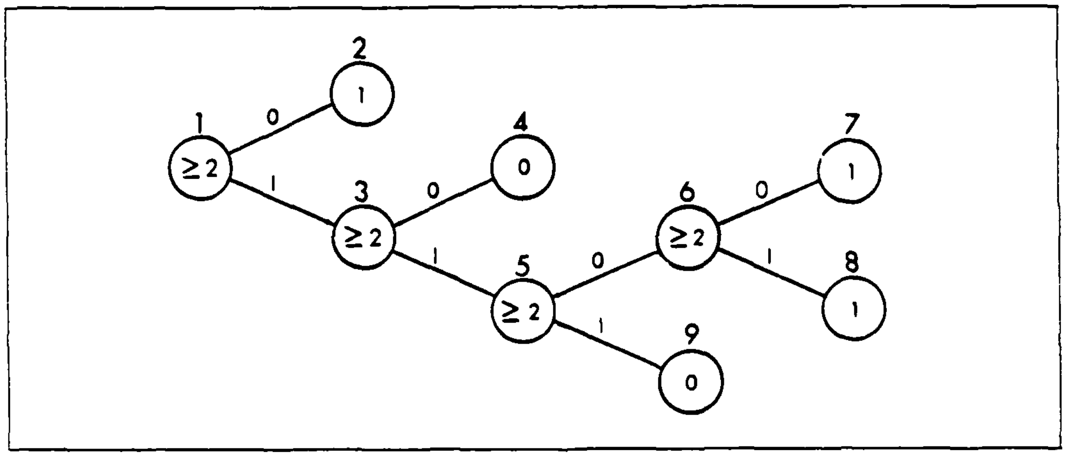


Figure 1.3 Tree Diagram for CCRA

The main interest is the length of the CRI, i.e., of the number of slots in the CRI, because system throughput is determined by dividing CRI into the number of packets involved in the original collision. Capetanakis proves that his scheme achieves throughputs above the 1/e "barrier" for slotted-ALOHA. The throughput for the Poisson source model and an infinite number of independent identical sources (users) is 0.346. He also proposed a dynamic CRA which leads to a maximum throughput of 0.430. A similar idea was proposed by J. F. Hayes [Ref. 7].

2. Massey's Algorithm

Observing that if a collision slot is followed by an empty slot, one slot can be saved by repeating the random retransmission (realized by probability 0.5) before a predictably certain collision is allowed. Massey [Ref. 8] improved the Capetanakis algorithm, called Modified Capetanakis Collision Resolution Algorithm (MCCRA). MCCRA is like the CCRA algorithm, except when the feedback indicates a slot in which a set of users who flipped 0 should retransmit is empty. Then each user involved in the most recent collision flips a binary fair coin. Those flipping 0 retransmit in the very next slot, those flipping 1 retransmit in the next slot after the collision (if any) among those flipping 0 is resolved. The following example shows how MCCRA works.

Figure 1.4 gives a binary tree for a CRI containing two packets in the original collision, and for which both users flipped 1 on the first two tosses of their binary coin, but one flipped a 0 and the other a 1 on their third toss. The nodes labelled "skip" and having no slot number written above them correspond to points where the feedback indicates that certain users should immediately flip their binary coins to thwart a certain collision. In this example, we need two fewer slots than CCRA because the nodes labelled "skip" would become collision slots [Ref. 8:p. 80].

MCCRA, unlike CCRA, requires the feedback information to distinguish between empty slots and the slots with one packet. Massey's modified algorithm (MCCRA) induces a throughput equal to 0.375 in the static case, and a throughput equal to 0.462 for the dynamic case. However, MCCRA algorithms are very sensitive to channel errors. If channel errors occur, the CRI never terminates and no packets are transmitted after some point.

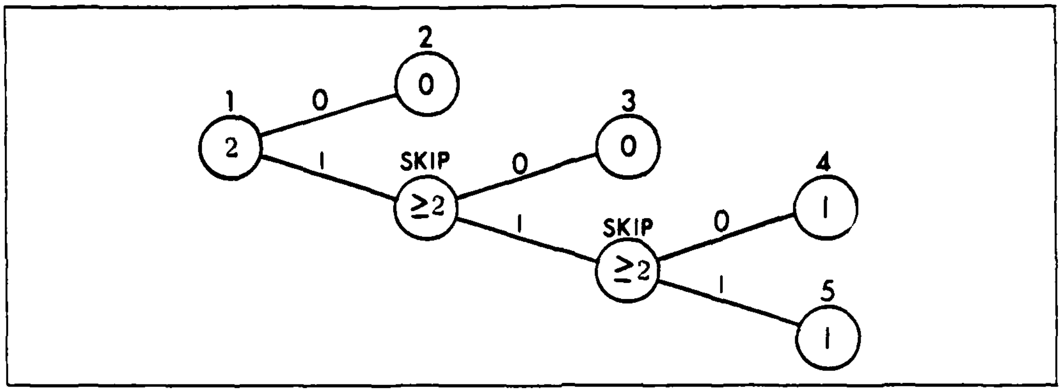


Figure 1.4 Tree Diagram for MCCRA

3. GPK Algorithm

In 1982, Georgiadis and Papantoni-Kazakos [Ref. 9] proposed a further improved CRA which is stable at higher input rate. Their algorithm involves the knowledge of multiplicity, i.e., the number of users involved in a transmission. At first, they assumed that the number of packets involved in a collision can always be detected correctly by a device called "energy detector". This protocol is called a Collision Resolution with Additional Information (CRAI). CRAI has a first-come-first-serve feature and can be described as follows.

Suppose n users are known to be involved in a collision. Then each of these users independently flips a bias coin with $P(H) = \sigma_n$ and $P(T) = 1-\sigma_n$ to break into two subgroups. Collision (if any) in the second group will not be resolved until all users in the first group have successfully transmitted. The optimal value of σ_n can be determined by classic dynamic programming.

Again, they proposed a modified version of CRAI (MCRAI) for the case in which the energy detector is

capable of telling the multiplicity of a transmission up to a certain limit. This MCRAI maintains the FCFS characteristics of CRAI and has certain robust properties. It means that MCRAI is robust in the presence of certain channel errors.

Georgiadis and Papantoni-Kazakos observed that CRAI stability region is obtained for an input rate below 0.5324. Stability region of MCRAI, for energy detectors of capacity 7, is very much the same as the stability region of CRAI.

Almost the same type of CRA was developed independently in the USSR by Tsybakov and Mikhailov [Ref. 11]. A slightly different conflict resolution algorithm was introduced by Gallager [Ref. 10] for the same model as in Capetanakis' and Massey's models. His algorithm also has stability and FCFS characteristics. Also, the algorithm realizes its maximum throughput at 0.4872. Gallager's protocol was later improved by Humblet [Ref. 12], with an increased throughput 0.48775.

C. SCOPE OF RESEARCH

As seen in previous sections, recent research on random access techniques has been centering around efficient collision resolution algorithms. So far, most of the work reported in the literature only consider single random accessed channel. Collision resolution algorithms applicable to a multichannel system have rarely been studied. The purpose of this thesis is to propose and analyze a few collision resolution protocols suitable for multichannel environment. For the multichannel CRA's, one can group them into the following categories.

Category 1 - The collision in each channel is resolved independently. New transmissions should not be attempted

until collision in all channels are cleared. At the beginning of a CRI, a station (user) with a packet to be transmitted randomly picks one channel and transmits. In selecting users for transmissions at the beginning of a CRI, Gallager's idea [Ref. 10] can be used. All ready users may also be allowed to transmit at the beginning of a CRI.

Category 2 - Each channel runs independently. At the beginning of the CRI of a certain channel, we can either allow all ready users to transmit or select users for transmission using Gallager's idea.

Category 3 - Collisions in all channels are resolved jointly by efficient algorithms. Otherwise it remains identical to the first category.

In this thesis, category 3 is emphasized, and some collision resolution protocols are proposed. It is further assumed that channel status information on the feedback channel is either ternary or m-ary. A ternary feedback specifies whether the channel is empty, successful, or colliding. An m-ary feedback adds the information about the number of users participating in a transmission. Under category 3, the following three collision resolution algorithms are studied. Detailed descriptions will be given in each corresponding chapter.

1. Static M-ary Splitting

At time t, each of the packets arriving during in "t_0, $t_0 + \Delta$" is transmitted through a randomly picked channel. Packets arriving in "$t_0 + \Delta$, $t_0 + 2\Delta$" will not be transmitted until any collisions of the previously transmitted packets are completely resolved. Energy detectors are not available. When collision is detected at a certain time slot, all packets involved in collision within this time slot are divided into m groups. Each packet

is randomly placed in one of m groups by flipping an m-faced dice. The packets in the first group are retransmited immediately through one of the randomly picked available channels. The second group is allowed to transmit after all of the packets in the first group are successfully transmitted, and so on.

2. Dynamic M-ary Splitting

An infinite energy detector is made available in the static m-ary splitting protocol. Therefore, the number of packets involved in a collision is known and we can accordingly determine the value of m (number of groups) which minimizes the CRI. The rest of the protocol remains the same as in static m-ary splitting.

3. Binary Splitting with Energy Detector

Assume the availability of an infinite energy detector. After a collision, each user involved in the collision randomly flips a biased coin to either join the first or the second group. Let σ_n denote the probability that a user will join the first group when the collision involves "n" users. Again, we shall optimize σ_n. To transmit a packet, one of the channels again will be picked randomly. If a user decides not to retransmit immediately, then its packet will be held until any collision involved in the previous slot is resolved.

D. OUTLINE OF THIS THESIS

In Chapter II, static m-ary splitting is analyzed and compared with simulation. The main results include:

a) The average number of slots, $\bar{L}_n$, required to resolve a collision involving "n" users is obtained. Through numerical examples, it is observed that binary

splitting achieves the minimum $\overline{L}_n$. The values of $\overline{L}_n$ are verified by computer simulation.

b) It is proved that $\overline{L}_n$ is linearly bounded. That is, there exist n_o, β_{ln_o}, β_{un_o} and C such that β_{ln_o}-C < $\overline{L}_n$ <β_{un_o}-C for all $n \geq n_o$.

c) Based on the linear bounding of $\overline{L}_n$, the maximum achievable throughput is obtained.

Dynamic m-ary splitting is considered in Chapter III. Results similar to those mentioned above are obtained. Dynamic m-ary splitting performs better than the best static m-ary splitting, i.e., the static binary splitting studied in Chapter II.

In Chapter IV, the binary splitting with energy detector protocol is analyzed. It is again observed that the dynamic m-ary splitting outperforms this protocol. Chapter V contains numerical results which allow comparisons among the performances of the protocols be made. Finally, in Chapter VI, we conclude this work and discussed some prospects for future study. This thesis also contains appendices which include listings of key programs used in computer simulation and some numerical data.

II. STATIC M-ARY SPLITTING

In this chapter, the study of the static m-ary splitting for resolving collisions in a multichannel system occurs. We focus on theoretical analysis of this protocol in this chapter and defer numerical examples together with discussions until Chapter V.

A. GENERAL OPERATION

In this section, the m-ary splitting algorithm is stated. We consider an infinite number of independent, identical, bursty packet transmitting users which collectively form a Poisson traffic model. Also errorless, time-slotted multichannel is assumed.

Let "t" be the current time and let stations, each of which generated a packet during the period $(t_0, t_0+\Delta)$ send their packets at time "t". Each of these stations randomly selects one of the α channels and sends its packet. Assume each user knows whether his transmission is a success or not immediately by listening to the channel after sending a packet. When collision is detected, colliding users involved in the current slot of all channels are divided into "m" groups by flipping an m-faced dice. The packets in the first group are retransmitted in the very next slot. The second group will be transmitted after collision, if any, among those in the first group has been cleared, then the third group, and so on.

B. DERIVATION OF $\bar{L}_N$

Let Y_{ij} denote the probability that j users succeed when i users attempt their retransmissions in a slot which contains α cells (channels). Of course $0 \le j \le i$. At this point, the following question arises:

Question: distribute i packets to α channels, what is the probability that j of the channels each contains exactly one packet while each of the others contain either zero or more than one packet ? If a channel slot contains only one packet, it will be called a success.

The answer can be found as follows: let X^{α}_{ij} denote the number of arrangements in distributing i users (packets) into α channels which results in j success, then

$$X^{\alpha}_{ij} = \begin{cases} {}_{\alpha}C_j \cdot {}_iC_j \cdot j! \; [(\alpha-j)^{i-j} - X^{\alpha-j}_{i-j,1} - \cdots - X^{\alpha-j}_{i-j,i-j}] \\ \qquad\qquad\qquad\qquad , \quad 0 \le j \le \min(\alpha, i)-1, \quad i \le \alpha \\ {}_{\alpha}C_j \cdot {}_iC_j \cdot j! \qquad , \quad j = \min(\alpha, i), \quad i \le \alpha \\ 0 \qquad\qquad\qquad\quad , \quad j = \min(\alpha, i), \quad i > \alpha \end{cases}$$

and

$$Y_{ij} = X^{\alpha}_{ij} \;/\; \alpha^i \qquad (2.1)$$

In (2.1), ${}_{\alpha}C_j$ is the binomial coefficient with parameters α and j. The factor ${}_{\alpha}C_j \cdot {}_iC_j \cdot j!$ is used to represent the number of arrangements in selecting j of α channels, j out of i packets, distributing j packets into j channels so that each of them is a success. The factor $(\alpha-j)^{i-j}$ ... then tells the number of arrangements in distributing the remaining $i \cdot j$ packets into the remaining α-j channels which results in no success at all. Notice that X^{α}_{ij} is expressed in recursive form. Numerical values of Y_{ij} will be discussed in Chapter V.

1. $\bar{L}_n$ of Binary Splitting (m=2)

Suppose that n packets have collided and $\bar{L}_n$ denotes the average number of slots required to solve this collision. In this case, the problem is equivalent to the following: after a collision is detected, the users involved randomly and independently flip a fair coin to determine whether to retransmit in the next slot or wait until later. Of course, in retransmiting a packet, one of the α channels will be picked randomly.

- Remark: when $\alpha = 1$, and if no one decides to retransmit immediately (in other words, if an empty slot is observed), Massey's idea [Ref. 7] is to restart the process immediately in order to avoid a guaranteed collision. However, when $\alpha > 1$, it seems unnecessary to follow Massey's idea.

Thus

$$\bar{L}_n = 2 + \sum_{i=0}^{n} P(n,i)\left[\sum_{j=0}^{\min(\alpha,n-i)} Y_{n-ij}L_{n-i-j} + \sum_{j=0}^{\min(\alpha,i)} Y_{ij}L_{i-j}\right] \quad (2.2)$$

The 2 in (2.2) results from the fact that each splitting requires at least two slots. In other words, for binary splitting, we need at least two slots to resolve a collision. Obviously, $L_0 = L_1 = 0$. Furthermore, P(n,i) is defined to be

$$P(n,i) = {}_nC_i\ (1/2)^n \quad (2.3)$$

which represents the probability that i of the n users decide to try again immediately. Substitute (2.3) into (2.2) we have

$$\bar{L}_n = 2 + \sum_{i=0}^{n} {}_nC_i\left(\tfrac{1}{2}\right)^n\left[\sum_{j=0}^{\min(\alpha,n-i)} Y_{n-ij}L_{n-i-j} + \sum_{j=0}^{\min(\alpha,i)} Y_{ij}L_{i-j}\right] \quad (2.4)$$

Through algebraic manipulation, we can obtain $\bar{L}_n$ in recursive form as follows:

$$\bar{L}_n = \frac{1}{1-2(1/2)^n Y_{n0}} \left[2+2(1/2)^n \sum_{j=1}^{\min(n,\alpha)} Y_{nj} L_{n-j} \right.$$

$$\left. + 2\sum_{j=0}^{n-1} {}_nC_i (1/2)^n \sum_{j=0}^{\min(\alpha,i)} Y_{ij} L_{i-j} \right] \tag{2.5}$$

when $n \geq 2$.

2. $\bar{L}_n$ of Ternary Splitting (m=3)

First define

$$P(n,i_1,i_2) = [n!/\{i_1!i_2!(n-i_1-i_2)!\}](1/3)^n \tag{2.6}$$

Physically $P(n,i_1,i_2)$ represents the probability of dividing n users into three groups of sizes i_1, i_2 and $n-i_1-i_2$. Proceed similarly to binary splitting, we have

$$\bar{L}_n = [1/\{1-3(1/3)^n Y_{n0}\}][3 + \sum_{\substack{i_1=0 \\ \max(i_1,i_2,n-i_1-i_2)<n}}^{n} \sum_{i_2=0}^{n-i_1} P(n,i_1,i_2)$$

$$\{\sum_{j_0=0}^{\min(\alpha,n-i_1-i_2)} Y_{n-i_1-i_2 j_0} L_{n-i_1-i_2-j_0}$$

$$+ \sum_{j_1=0}^{\min(\alpha,i_1)} Y_{i_1 j_1} L_{i_1-j_1} + \sum_{j_2=0}^{\min(\alpha,i_2)} Y_{i_2 j_2} L_{i_2-j_2} \}] \tag{2.7}$$

3. $\bar{L}_n$ of General M-ary Splitting

In general, we define

$$P(n;\underline{i}) \triangleq P(n;i_1,\ldots,i_m)$$

$$= [n!/(i_1!i_2!\ldots i_m!)](1/m)^n \tag{2.8}$$

where $i_l \geq 0$ and $\sum_{l=1}^{m} i_l = n$

then

$$\overline{L}_n = m + \sum_{\substack{i_1+\cdots+i_m=n \\ i_\ell \geq 0}} P(n;\underline{i})\left[\sum_{\ell=1}^{m} \sum_{j_\ell=0}^{\min(\alpha,i_\ell)} Y_{i_\ell j_\ell} L_{i_\ell - j_\ell}\right] \tag{2.9}$$

From (2.9), we obtain

$$\overline{L}_n = \frac{1}{1-m(1/m)^n Y_{n0}}\left[m + m(1/m)^n \sum_{j=1}^{\min(n,\alpha)} Y_{nj}L_{n-j} \right.$$

$$\left. + \sum_{\substack{i_1+\cdots+i_m=n \\ i_\ell > 0 \\ \max(i_1\cdots i_m)<n}} P(n;\underline{i}) \sum_{\ell=1}^{m} \sum_{j_\ell=1}^{\min(\alpha,i_\ell)} Y_{i_\ell j_\ell} L_{i_\ell - j_\ell}\right] \tag{2.10}$$

and $L_0 = L_1 = 0$. Notice that (2.10) is expressed in recursive form.

The result of L_n for general m-ary splitting can be further simplified as follows:

$$\overline{L}_n = m + m\sum_{i=0}^{n-1} \binom{n}{i}\left(\frac{1}{m}\right)^i\left(1-\frac{1}{m}\right)^{n-i} \sum_{j=0}^{\min(\alpha,i)} Y_{ij}L_{i-j}$$

$$+ m(1/m)^n \sum_{j=1}^{\min(\alpha,n)} Y_{nj}L_{n-j} + m(1/m)^n Y_{n0}L_n \tag{2.11}$$

Finally, from (2.11) we obtain

$$\overline{L}_n = \frac{1}{1-m(1/m)^n Y_{n0}}\left[m + m(1/m)^n \sum_{j=1}^{\min(\alpha,n)} Y_{nj}L_{n-j}\right.$$

$$\left. + m\sum_{i=0}^{n-1} \binom{n}{i}\left(\frac{1}{m}\right)^i\left(1-\frac{1}{m}\right)^{n-i} \sum_{j=0}^{\min(\alpha,i)} Y_{ij}L_{i-j}\right] \tag{2.12}$$

The form shown in (2.12) is very convenient for numerical calculations.

The following example helps verify the correctness of (2.12). In this example we consider α=2, m=2 and n=2. Figure 2.1 shows the state transition diagram for this example. States such as (1,1) represents the splitting of

two users into two groups of sizes 1 and 1. From Figure 2.1, $\overline{L}_2$ can be obtained as follows:

$$L_2(z) = (0.25+0.25)z^2 0.5 + (0.25+0.25)z^2 0.5\ L_2(z) + 0.5z^2$$
$$= [1/(1-0.25z^2)](0.75z^2)$$

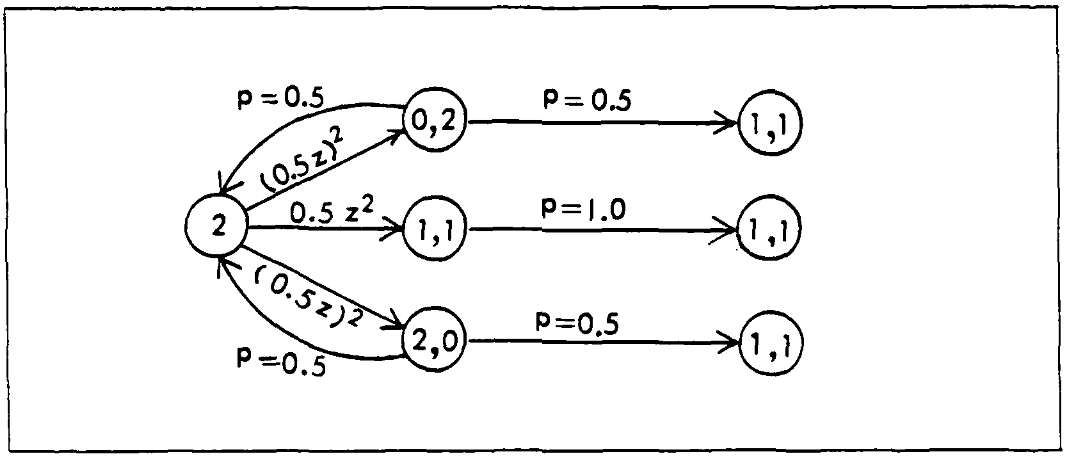


Figure 2.1 State Transition Diagram
of Binary Splitting with N=2

Differentiate $L_2(z)$ obtained above and set $z = 1$, we obtain

$$\overline{L}_2 = [(0.75)(1.5) + (0.75)(0.5)]/(0.75)^2$$
$$= 8/3$$

While from (2.12), we have

$$\overline{L}_2 = \frac{1}{1-Y_{20}2(1/2)^2}[2 + 2(0.5)^2(Y_{21}L_1 + Y_{22}L_0) + 0]$$
$$= 8/3$$

Thus we have the agreement.

C. $\overline{L}_N^D$

We use $\overline{L}_n^d$ to denote the actual average number of slots required in successfully transmitting a total of n packets including the initial slots which contain collisions (if any). Then

$$\overline{L}_n^d = 1 + \sum_{j=0}^{n} Y_{nj} L_{n-j} \tag{2.13}$$

This expression is valid for all the protocols considered in this study.

D. LINEAR BOUNDING OF $\overline{L}_N$

By linear bounding, we mean we assume there exist n_o, β_{1n_o}, β_{un_o} and C such that $\overline{L}_n$ satisfies

$$U(n_o-n)L_n + U(n-n_o+1)(\beta_{1n_o}n-C) \le \overline{L}_n$$

$$\le U(n_o-n)L_n + U(n-n_o+1)(\beta_{un_o}n-C) \tag{2.14}$$

where $U(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$

In practice it is reasonable to assume $n_o > \alpha$. Then

$$\overline{L}_n \le \frac{1}{1-m(1/m)^n Y_{n0}} [\, m + m(1/m)^n \sum_{j=1}^{\min(\alpha,n)} Y_{nj} \{U(n_o-n+j)L_{n-j}$$

$$+ U(n-j-n_o+1)(\beta_{un_o}(n-j)-C)\}$$

$$+ m \sum_{i=0}^{n-1} \binom{n}{i} \left(\frac{1}{m}\right)^i \left(1-\frac{1}{m}\right)^{n-i} \sum_{j=0}^{\min(\alpha,i)} Y_{nj} \{U(n_o-i+j)L_{i-j}$$

$$+ U(i-j-n_o+1)(\beta_{un_o}(i-j)-C)\}] \tag{2.15}$$

Since $U(n_o-n+j)+U(n-j-n_o+1) = 1$, we have

$$\bar{L}_n \le \frac{1}{1-m(1/m)^n Y_{n0}} [m+m(1/m)^n \sum_{j=1}^{\min(n,\alpha)} Y_{nj}\{U(n_o-n+j)(L_{n-j}+C)$$

$$+ \beta_{un_o}(n-j)-U(n_o-n+j)\beta_{un_o}(n-j)-C\}$$

$$+ m\sum_{i=0}^{n-1} \binom{n}{i}\left(\frac{1}{m}\right)^i\left(1-\frac{1}{m}\right)^{n-i} \sum_{j=0}^{\min(\alpha,i)} Y_{ij}\{U(n_o-i+j)(L_{i-j}+C)$$

$$+ \beta_{un_o}(i-j)-U(n_o-i+j)\beta_{un_o}(i-j)-C\}] \qquad (2.16)$$

In (2.16), the terms which include C can be arranged to become

$$m(1/m)^n \sum_{j=1}^{\min(n,\alpha)} Y_{nj}+m\sum_{i=0}^{n-1}\binom{n}{i}(1/m)^i(1-1/m)^{n-i}\sum_{j=0}^{\min(\alpha,i)} Y_{ij}$$

$$= m - m(1/m)^n\, Y_{n0} \qquad (2.17)$$

Similarly, terms containing β_{un_o} can be combined to obtain

$$m(1/m)^n \sum_{j=1}^{\min(n,\alpha)} Y_{nj}(n-j)$$

$$+ m\sum_{i=0}^{n-1}\binom{n}{i}(1/m)^i(1-1/m)^{n-i}\sum_{j=0}^{\min(i,\alpha)} Y_{ij}(i-j) \qquad (2.18)$$

Since

$$\sum_{j=1}^{\alpha} Y_{nj} = 1 - Y_{n0}$$

we have

$$\sum_{j=1}^{\alpha} Y_{nj}(n-j) = n(1-Y_{n0}) - \sum_{j=0}^{\alpha} jY_{nj}$$

Let us now obtain the closed-form representation of

$$\sum_{j=1}^{\alpha} jY_{nj}$$

The result can be obtained by answering the follow question. Given n packets and α channels such that $n \geq \alpha$. Distribute these packets into the channels in a purely random manner. What is the expected number of channels, each of them containing exactly one packet, i.e., $\sum_{j=0}^{\alpha} jY_{nj}$? Define

$$X_i = \begin{cases} 1 & \text{if channel i contains exactly one packet} \\ 0 & \text{otherwise} \end{cases}$$

then

$$P(x_i = 1) = n(1/\alpha)(1-1/\alpha)^{n-1}$$

therefore

$$\sum_{j=0}^{\alpha} jY_{nj} = \sum_{i=1}^{\alpha} E(X_i) = n(1-1/\alpha)^{n-1} \qquad (2.19)$$

As a matter of fact, this result is good not only for $n \geq \alpha$, but also for $n < \alpha$, since

$$\sum_{j=0}^{n} jY_{nj} = \alpha n(1/\alpha)(1-1/\alpha)^{n-1}$$

$$= n(1-1/\alpha)^{n-1}$$

Thus (2.18) now becomes

$$m[\sum_{i=0}^{n-1} \binom{n}{i}(1/m)^i(1-1/m)^{n-i}\sum_{j=0}^{\min(\alpha,i)} Y_{ij}(i-j)-(1/m)^n Y_{n0}n]$$

$$= n-m(\tfrac{1}{m})^n Y_{n0}n-m \sum_{i=0}^{n} \binom{n}{i}(\tfrac{1}{m})^i(1-\tfrac{1}{m})^{n-i}i(1-1/\alpha)^{i-1} \qquad (2.20)$$

The second term of (2.20) can be reduced to

$$m \sum_{i=0}^{n} \binom{n}{i}(1/m)^i(1-1/m)^{n-i}i(1-1/\alpha)^{i-1} = n(1-1/m\alpha)^{n-1}$$

therefore (2.20) now can be written as

$$n[1-m(1/m)^n Y_{n0}] - n(1-1/m\alpha)^{n-1} \tag{2.21}$$

Subsitute (2.21) into (2.15) and apply $n \geq n$, after tedious algebraic manipulation, we obtain

$$\begin{aligned}
\overline{L}_n \leq\ & \beta_{un_o} n - C + [1/\{1-m(1/m)^n Y_{n0}\}][\ m - (m-1)C \\
& - \beta_{un_o}\{n(1-1/m\alpha)^{n-1} + m(1/m)^n \sum_{j=1}^{\min(\alpha,n)} Y_{nj} U(n_o-n+j)(n-j) \\
& + m \sum_{i=0}^{n-1} \binom{n}{i} (1/m)^i (1-1/m)^{n-i} \sum_{j=0}^{\min(\alpha,i)} Y_{nj} U(n_o-i+j)(i-j)\} \\
& + \{m(1/m)^n \sum_{j=1}^{\min(\alpha,n)} Y_{nj} U(n_o-n+j)(L_{n-j}+C) \\
& + m \sum_{i=0}^{n-1} \binom{n}{i} (\tfrac{1}{m})^i (1-\tfrac{1}{m})^{n-i} \sum_{j=0}^{\min(\alpha,i)} Y_{ij} U(n_o-i+j)(L_{i-j}+C)\}]
\end{aligned} \tag{2.22}$$

Thus

$$\beta_{un_o} = \sup_{n \geq n_o} [\ A\ /\ B\ ] \tag{2.23}$$

where $A = m - (m-1)C$

$$+ m \sum_{i=0}^{n} \binom{n}{i} (\tfrac{1}{m})^i (1-\tfrac{1}{m})^{n-i} \sum_{j=0}^{\min(\alpha,i)} Y_{ij} U(n_o-i+j)(L_{i-j}+C)$$

$$B = n(1-\tfrac{1}{m\alpha})^{n-1}$$

$$+ m \sum_{i=0}^{n} \binom{n}{i} (\tfrac{1}{m})^i (1-\tfrac{1}{m})^{n-i} \sum_{j=0}^{\min(\alpha,i)} Y_{ij} U(n_o-i+j)(i-j)$$

Also, we have

$$\beta_{ln_o} = \inf_{n \geq n_o} [\ A\ /\ B\ ] \tag{2.24}$$

Now we need to select a proper value for C. The selection of C can be done as follows. For sufficiently large n, each m-ary splitting results in m groups. Because collided users in each group are resolved collectively and each group has approximately n/m users, we have

$$\overline{L}_n \cong m + m\, L_{n/m}$$

- claim : $\overline{L}_n \cong \beta n - m/(m-1)$
- check : $m + m\, L_{n/m} = m + m[\beta(n/m) - m/(m-1)]$
  $= \beta n - m/(m-1)$

thus for m-ary splitting, C is

$$C = m/(m-1) \tag{2.25}$$

If we substitute (2.25) into (2.23), then the first term in the numerator of (2.23) will be $m - (m-1)C = 0$.

E. THROUGHPUT ANALYSIS

In this section, maximal achievable throughput λ_{max} is obtained. For this purpose, use the results obtained in the preceding section we have

$$\beta_{ln_o} n - C \le \overline{L}_n \le \beta_{un_o} n - C \quad \forall \; n \ge n_o > \alpha \tag{2.26}$$

We can generalize (2.26) to

$$\beta_{ln_o} n - C + \sum_{i=0}^{n_o-1} C_{i_l} \delta_{i_n} \le \overline{L}_n$$

$$\le \beta_{un_o} n - C + \sum_{i=0}^{n_o-1} C_{i_u} \delta_{i_n} \quad \forall \; n \ge 0 \tag{2.27}$$

where $C_{i_l} = \overline{L}_i - \beta_{ln_o} i + C$

$C_{i_u} = \overline{L}_i - \beta_{un_o} i + C$

Equation (2.13) can be modified for this protocol as follows

$$\bar{L}_n^d = 1 + \sum_{j=0}^{\min(\alpha,n)} Y_{nj} L_{n-j} \tag{2.28}$$

Subsituting (2.28) into (2.27) gives

$$1 + \sum_{j=0}^{\min(\alpha,n)} Y_{nj}[\beta_{1n_o}(n-j) - C + \sum_{i=0}^{n_o-1} C_{i_1}\delta_{i,n-j}] \le \bar{L}_n^d$$

$$\le 1 + \sum_{j=0}^{\min(n,\alpha)} Y_{nj}[\beta_{un_o}(n-j) - C + \sum_{i=0}^{n_o-1} C_{i_u}\ \delta_{i,n-j}] \tag{2.29}$$

It can be proven that

$$\sum_{j=0}^{\min(\alpha,n)} Y_{nj}\beta_{un_o}(n-j) = \beta_{un_o}[n - n(1-1/\alpha)^{n-1}] \tag{2.30}$$

where $n(1-1/\alpha)^{n-1}$ is only good for $\alpha \ge 2$. Also

$$\sum_{j=0}^{\min(\alpha,n)} Y_{nj}\ C = C$$

and

$$\sum_{j=0}^{\min(\alpha,n)} Y_{nj} \sum_{i=0}^{n_o-1} C_{i_u}\delta_{i,n-j}$$

Let the mean arrival rate of the Poisson process be λ, and N denote the random variable representing the number of packets arrived during $(t_0\ ,\ t_0 + \Delta)$, we clearly have

$$E[N] = \lambda\Delta$$

$$P[N = n] = e^{-\lambda\Delta}(\lambda\Delta)^n\ /\ n!$$

Now take the expected value for all parts in (2.29). From (2.30) we have

$$\sum_{n=0}^{x} n(1-1/\alpha)^{n-1} e^{-\lambda\Delta} (\lambda\Delta)^n/n!$$

$$= \frac{d}{dz}\left[\sum_{n=0}^{\infty} e^{-\lambda\Delta} (\lambda\Delta z)^n/n!\right]_{z=1-1/\alpha}$$

$$= \lambda\Delta\ e^{-\lambda\Delta} \tag{2.31}$$

next we have

$$\sum_{n=0}^{\infty} \Big[\sum_{j=0}^{\min(\alpha,n)} Y_{nj} \sum_{i=0}^{n_o-1} C_{i_u}\delta_{i,n-j}\Big]\ e^{-\lambda\Delta}(\lambda\Delta)^n/n!$$

$$= \sum_{n=0}^{n_o-1} \Big[\sum_{j=0}^{\min(\alpha,n)} Y_{nj} \sum_{i=0}^{n_o-1} C_{i_u}\delta_{i,n-j}\Big]\ e^{-\lambda\Delta}(\lambda\Delta)^n/n!$$

$$+ \sum_{n=n_o}^{n_o+\alpha-1} \Big[\sum_{j=n-n_o+1}^{\alpha} Y_{nj} \sum_{i=0}^{n_o-1} C_{i_u}\ \delta_{i,n-j}\Big]\ e^{-\lambda\Delta}(\lambda\Delta)^n/n! \tag{2.32}$$

Let us now define

$$L1 = \sum_{j=0}^{\min(\alpha,n)} Y_{nj} \sum_{i=0}^{n_o-1} C_{i_l}\delta_{i,n-j}$$

$$L2 = \sum_{j=n-n_o+1}^{\alpha} Y_{nj} \sum_{i=0}^{n_o-1} C_{i_l}\delta_{i,n-j}$$

$$U1 = \sum_{j=0}^{\min(\alpha,n)} Y_{nj} \sum_{i=0}^{n_o-1} C_{i_u}\delta_{i,n-j}$$

$$U2 = \sum_{j=n-n_o+1}^{\alpha} Y_{nj} \sum_{i=0}^{n_o-1} C_{i_u}\delta_{i,n-j} \quad \text{and} \quad x = \lambda\Delta$$

Then (2.29) implies

$$1 - C + \beta_{ln_o} x(1-e^{-x/\alpha}) + \sum_{n=0}^{n_o-1} L1\ e^{-x}x^n/n!$$

$$+ \sum_{n=n_o}^{n_o+\alpha-1} L2\ e^{-x}x^n/n! \le E[\bar{L}_n^d]$$

$$\le 1 - C + \beta_{un_o} x(1-e^{-x/\alpha})$$

$$+ \sum_{n=0}^{n_o-1} U1\ e^{-x}x^n/n! + \sum_{n=n_o}^{n_o+\alpha-1} U2\ e^{-x}x^n/n! \tag{2.33}$$

Clearly, it is sufficient that $E[\bar{L}_n^d] \leq \Delta$ guarantees the stability of the protocol. On the other hand, if $E[\bar{L}_n^d] > \Delta$, the protocol is definitely unstable. Then due to the upper bound in (2.33), we also conclude that the following condition is sufficient for stability:

$$1 - C + \beta_{un_o} x(1-e^{-x/\alpha}) + \sum_{n=0}^{n_o-1} U1\ e^{-x}x^n/n! + \sum_{n=n_o}^{n_o+\alpha-1} U2\ e^{-x}x^n/n! \leq \Delta \tag{2.34}$$

From (2.33) and (2.34), we obtain the following in a straightforward manner. For stability of the m-ary splitting, it is sufficient that the input rate λ satisfies the following condition:

$$\lambda \leq [x/\{1-C+\beta_{un_o} x(1-e^{-x/\alpha}) + \sum_{n=0}^{n_o-1} U1\ e^{-x}x^n/n! + \sum_{n=n_o}^{n_o+\alpha-1} U2\ e^{-x}x^n/n!\}] \tag{2.35}$$

Then

$$\lambda_{max} = \sup_x [\ x\ /\ [1-C+\beta_{un_o} x(1-e^{-x/\alpha}) + \sum_{n=0}^{n_o-1} U1\ e^{-x}x^n/n! + \sum_{n=n_o}^{n_o+\alpha-1} U2\ e^{-x}x^n/n!\}] \tag{2.36}$$

The maximization of (2.36) is straightforward, and provides us with the maximum achievable throughput, as well as the optimal Δ choice for the m-ary splitting. Indeed, maxima is obtained at $x \rightarrow \infty$, or equivalently at $\Delta \rightarrow \infty$.

III. DYNAMIC M-ARY SPLITTING

In this chapter dynamic m-ary splitting is considered for resolving collisions in a multichannel system. For dynamic m-ary splitting, we will assume that the existence of an infinite energy detector. Again we focus on theoretical analysis and defer numerical results until Chapter V.

A. DERIVATION OF $\overline{L}_N$

As mentioned before, dynamic m-ary splitting can be analyzed in the same state as static m-ary splitting except the application of an infinite energy detector. Upon the availability of an infinite energy detector, we can have $\overline{L}_n$ for this protocol as follows:

$$\begin{aligned}\overline{L}_n &= [1/\{1-m(n)(1/m(n))^n Y_{n0}\}][\, m(n) \\ &+ m(n)(1/m(n))^n \sum_{j=1}^{\min(\alpha,n)} Y_{nj} L_{n-j} \\ &+ m(n) \sum_{i=0}^{n-1} \binom{n}{i} \left(\frac{1}{m(n)}\right)^i \left(1-\frac{1}{m(n)}\right)^{n-i} \sum_{j=0}^{\min(\alpha,i)} Y_{ij} L_{i-j} \,]\end{aligned} \qquad (3.1)$$

where m is a function of n. One possible selection of m(n) is

$$m(n) = [\, n \,/\, \alpha \,]$$

At this expression, [x] means the smallest integer ≥ x.

B. LINEAR BOUNDING OF $\overline{L}_N$

Regarding the linear bounding of $\overline{L}_n$ of dynamic m-ary splitting in (3.1), they can be obtained from (2.23) by subsituting

$$m(n) \rightarrow m$$

$$c(n) = m(n)/\{m(n)-1\} \rightarrow C$$

Thus, we have

$$\begin{aligned}
\beta_{un_o} = \sup_{n \geq n_o} &[\{m(n)-(m(n)-1)C \\
&+ m(n) \sum_{i=0}^{n} \binom{n}{i} \left(\frac{1}{m(n)}\right)^{i} \left(1-\frac{1}{m(n)}\right)^{n-i} \sum_{j=0}^{\min(\alpha,i)} Y_{ij} U(n_o-i+j)(L_{i-j}+C)\} \\
&/ \{n\left(1-\frac{1}{m(n)\alpha}\right)^{n-i} \\
&+ m(n) \sum_{i=0}^{n} \binom{n}{i} \left(\frac{1}{m(n)}\right)^{i} \left(1-\frac{1}{m(n)}\right)^{n-i} \sum_{j=0}^{\min(\alpha,i)} Y_{ij} U(n_o-i+j)(i-j)\}]
\end{aligned} \qquad (3.2)$$

C. THROUGHPUT ANALYSIS

In this section, maximal achievable throughput λ_{max} of dynamic m-ary splitting is obtained. Using the same procedure as in Chapter II, we have

$$\beta_{1n_o} n - C(n) \leq \overline{L}_n \leq \beta_{un_o} n - C(n) \quad \forall \; n \geq n_o > \alpha \qquad (3.3)$$

As far as $\overline{L}_n^d$ is concerned for dynamic m-ary splitting, we can use the same expression as in (2.28). Then we have

$$E[\overline{L}_n^d] = \sum_{n=0}^{\infty} \overline{L}_n^d \, P\{N = n\} \qquad (3.4)$$

Substitute (2.28) into (3.4) we obtain

$$E[\bar{L}_n^d] \le 1 + \beta_{un_o}\lambda\Delta(1-e^{-\lambda\Delta/\alpha})P\{N=n\}$$
$$+ \sum_{n=0}^{n_o-1} [\sum_{j=0}^{\min(n,\alpha)} Y_{nj}(\bar{L}_{n-j}-\beta_{un_o}(n-j))]P\{N=n\}$$
$$+ \sum_{n=n_o}^{n_o+\alpha-1} [\sum_{j=n-n_o+1}^{\alpha} Y_{nj}(\bar{L}_{n-j}-\beta_{un_o}(n-j))]P\{N=n\}$$
$$- \sum_{n=n_o}^{n_o+\alpha-1} [\sum_{j=0}^{n-n_o} Y_{nj}C(n-j)]P\{N=n\}$$
$$- \sum_{n=n_o+\alpha-1}^{\infty} [\sum_{j=0}^{\alpha} Y_{nj}C(n-j)]P\{N=n\} \qquad (3.5)$$

Defining, the possible values of C(n), if $n>\alpha$:

$$C(n) \ge 1$$

and substituting the same expression of P[N = n] as in previous chapters, we have

$$E[\bar{L}_n^d] \le \beta_{un_o}\lambda\Delta(1-e^{-\lambda\Delta/\alpha})$$
$$+ \sum_{n=0}^{n_o-1} [\sum_{j=0}^{\min(n,\alpha)} Y_{nj}(1+\bar{L}_{n-j}-\beta_{un_o}(n-j))]\ e^{-\lambda\Delta}(\lambda\Delta)^n/n!$$
$$+ \sum_{n=n_o}^{n_o+\alpha-1} [\sum_{j=n-n_o+1}^{\infty} Y_{nj}(1+\bar{L}_{n-j}-\beta_{un_o}(n-j))]\ e^{-\lambda\Delta}(\lambda\Delta)^n/n! \qquad (3.6)$$

Similar to Chapter II, $E[\bar{L}_n^d] \le \Delta$ guarantees system stability. Thus,

$$\beta_{un_o}\lambda\Delta(1-e^{-\lambda\Delta/\alpha})$$
$$+ \sum_{n=0}^{n_o-1} [\sum_{j=0}^{\min(\alpha,n)} Y_{nj}(1+\bar{L}_{n-j}-\beta_{un_o}(n-j))]e^{-\lambda\Delta}(\lambda\Delta)^n/n!$$
$$+ \sum_{n=0}^{n_o+\alpha-1} [\sum_{j=n-n_o+1}^{\alpha} Y_{nj}(1+\bar{L}_{n-j}+\beta_{un_o}(n-j))]e^{-\lambda\Delta}(\lambda\Delta)^n/n! \le \Delta \qquad (3.7)$$

With algebraic manipulation and define $x = \lambda\Delta$. Finally, we have

$$\lambda_{max} = \sup_{x} \Big\{ x \Big/ \Big[\beta_{un_o} x(1-e^{-x/\alpha})$$
$$+ \sum_{n=0}^{n_o-1} \{ \sum_{j=0}^{\min(\alpha,n)} Y_{nj}(1+\bar{L}_{n-j}-\beta_{un_o}(n-j)\} e^{-x} x^n/n!$$
$$+ \sum_{n=n_o}^{n_o+\alpha-1} \{ \sum_{j=n-n_o+1}^{\alpha} Y_{nj}(1+\bar{L}_{n-j}-\beta_{un_o}(n-j)\} e^{-x} x^n/n!\Big]\Big\} \qquad (3.8)$$

As a matter of fact, by comparing (2.36) and (3.8), we observe that (3.8) is the result of (2.36) with $C = 1$.

IV. BINARY SPLITTING WITH ENERGY DETECTOR

Through numerical examples to be given in Chapter V, we shall see that the best static m-ary splitting is binary splitting. Therefore, in this chapter we study the binary splitting with energy detector protocol. Only theoretical analysis of this protocol is focused in this chapter and numerical values and discussions are deferred until the next chapter.

A. GENERAL OPERATION

As stated in Chapter I, we assume the availability of infinite energy detectors. After a collision is detected, "n" users involved in a collision divided into two groups by flipping a biased coin. We shall find σ_n, the optimal probability that a user will join the first group, which minimizes $\overline{L}_n$. To transmit a packet, one of the available channels will be picked randomly. The users involved in the second group, decide not to retransmit immediately, will be held until collisions, if any, among those in the first group are cleared and then transmit.

B. DERIVATION OF $\overline{L}_N$

We will use the same Y_{ij} of (2.1). With this value of Y_{ij}, $\overline{L}_n$ can be obtained

$$\overline{L}_n = 2 + \sum_{i=0}^{n} {}_nC_i \sigma_n{}^i (1-\sigma_n)^{n-i} [\sum_{j=0}^{\min(\alpha,i)} Y_{ij} L_{i-j} + \sum_{j=0}^{\min(\alpha,n-i)} Y_{n-ij} L_{n-i-j}] \tag{4.1}$$

where $n \geq 2$ and $L_0 = L_1 = 0$. The 2 in (4.1) results from the same fact stated on p. 26 . Equation (4.1) can be rewritten as follows

$$\overline{L}_n = [1/\{1-Y_{n0}(1-\sigma_n)^n-Y_{n0}\sigma_n^n\}][2+(\sigma_n^n+(1-\sigma_n)^n)\sum_{j=1}^{\min(\alpha,n)} Y_{nj}L_{n-j}$$
$$+\sum_{j=1}^{n-1}\{{}_nC_i\sigma_n^i(1-\sigma_n)^{n-i}+{}_nC_{n-i}\sigma_n^{n-i}(1-\sigma_n)^i\}\sum_{j=0}^{\min(\alpha,i)} Y_{ij}L_{i-j}] \qquad (4.2)$$

In (4.2), we assume that σ_n and $\overline{L}_n$ have been optimal.

If we adopt Massey's idea of MCCRA [Ref. 8] in this protocol, i.e., in each binary splitting, if the first slot is observed to be empty or all users collide again, then we have

$$\overline{L}_n = [1/\{1-Y_{n0}\sigma_n^n-(1-\sigma_n)^n\}][1$$
$$+\sigma_n^n\sum_{j=1}^{\min(\alpha,n)} Y_{nj}L_{n-j}+{}_nC_{n-1}\sigma_n^{n-1}(1-\sigma_n)$$
$$+\sum_{i=1}^{n-1}\{{}_nC_i\sigma_n^i(1-\sigma_n)^{n-i}+{}_nC_{n-i}\sigma_n^{n-i}(1-\sigma_n)^i\}\sum_{j=0}^{\min(\alpha,i)} Y_{ij}L_{i-j}] \qquad (4.3)$$

This method shall be referred to as the modified binary splitting with energy detector. As in MCCRA, $\overline{L}_n$ can be reduced remarkably by using (4.3). Regarding of $\overline{L}_n^d$, we can use same the expression as in (2.13) for this protocol.

C. LINEAR BOUNDING OF $\overline{L}_N$

We will consider the linear bounding of $\overline{L}_n$, i.e.,

$$\beta_{ln_o}n - C \le \overline{L}_n \le \beta_{un_o}n - C$$

Since α , the number of available channels, cannot be too big for practical interest we assume $n_o>\alpha$, and define

$$P(n,i,\sigma_n) = {}_nC_i\ \sigma_n^i(1-\sigma_n)^{n-i} \qquad (4.4)$$

We shall consider the following two cases.

1. $\underline{N_o \le N \le N_o + a}$

For this case we obtain the expression of $\overline{L_n}$ as follows:

$$\overline{L_n} = [1/\{1-Y_{n0}(1-\sigma_n)^n - Y_{n0}\sigma_n^n\}] \Big\{ 2$$
$$+[(1-\sigma_n)^n+\sigma_n^n] \sum_{j=1}^{n-n_o} Y_{nj}L_{n-j}$$
$$+[(1-\sigma_n)^n+\sigma_n^n] \sum_{j=n-n_o+1}^{\alpha} Y_{nj}L_{n-j}$$
$$+\sum_{i=1}^{\alpha-1} W \sum_{j=0}^{\alpha} Y_{ij}L_{i-j} + \sum_{i=\alpha}^{n_o-1} W \sum_{j=0}^{\alpha} Y_{ij}L_{i-j}$$
$$+\sum_{i=n_o}^{n-1} W \Big[\sum_{j=0}^{i-n_o} Y_{ij}L_{i-j} + \sum_{j=i-n_o+1}^{\alpha} Y_{ij}L_{i-j}\Big] \Big\} \tag{4.5}$$

where W is defined as follows

$$W = P(n,i,\sigma_n) + P(n,n-i,\sigma_n)$$

and substitute (2.26) into (4.5), we have

$$\overline{L_n} \le [1/\{1-Y_{n0}(1-\sigma_n)^n - Y_{n0}\sigma_n^n\}] \Big\{ 2$$
$$+ [\sigma_n^n+(1-\sigma_n)^n] \sum_{j=1}^{n-n_o} Y_{nj}[\beta_{un_o}(n-j)-C]$$
$$+ [\sigma_n^n+(1-\sigma_n)^n \sum_{j=n-n_o+1}^{\alpha} Y_{nj}L_{n-j}$$
$$+\sum_{i=1}^{\alpha-1} W \sum_{j=0}^{i} Y_{ij}L_{i-j} + \sum_{i=2}^{n_o-1} W \sum_{j=0}^{\alpha} Y_{ij}L_{i-j}$$
$$+\sum_{i=n_o}^{n-1} W\Big[\sum_{j=0}^{i-n_o} Y_{ij}(\beta_{un_o}(i-j)-C) + \sum_{j=i-n_o+1}^{\alpha} Y_{ij}L_{i-j}\Big] \Big\} \tag{4.6}$$

Substitute the same expression for $\sum_{j=1}^{\alpha} Y_{nj}(n-j)$ and $\sum_{j=1}^{\alpha} Y_{nj}$ as in Chapter II and apply (2.19), after tedious algebraic manipulation, we obtain

$$\overline{L}_n \le \beta_{un_o} n - C + [1/\{1-Y_{n0}\sigma_n^n - Y_{n0}(1-\sigma_n)^n\}][A-B] \qquad (4.7)$$

where
$$A = 1+\sigma_n^n+(1-\sigma_n)^n+[\sigma_n^n+(1-\sigma_n)^n]\sum_{i=n-n_o+1}^{\alpha} Y_{nj}(L_{n-j}+C)$$
$$+\sum_{i=1}^{n_o-1} W \sum_{j=0}^{\min(\alpha,i)} Y_{ij}(L_{i-j}+C)$$
$$+\sum_{i=n_o}^{n-1} W \sum_{j=i-n_o+1}^{\alpha} Y_{ij}(L_{i-j}+C)$$

$$B = n[\sigma_n(1-\sigma_n/\alpha)^{n-1}+(1-\sigma_n)\{1-(1-\sigma_n)/\alpha\}^{n-1}]$$
$$+[\sigma_n^n+(1-\sigma_n)^n]\sum_{j=n-n_o+1}^{\alpha} Y_{nj}(n-j)$$
$$+\sum_{i=1}^{n_o-1} W \sum_{j=0}^{\min(\alpha,i)} Y_{ij}(i-j)$$

From this expression, we obtain the upper bound and lower bound of L_n,

$$\beta_{un_o} = \sup_{n_o \le n \le n_o+\alpha} [A/B]$$

$$\beta_{ln_o} = \inf_{n_o \le n \le n_o+\alpha} [A/B] \qquad (4.8)$$

2. <u>$N > N_O + \alpha$</u>

Using defined W, $\overline{L}_n$ for this case can be described as follows:

$$\overline{L}_n = [1/\{1-Y_{n0}(1-\sigma_n)^n - Y_{n0}\sigma_n^n\}] \Big\{ 2$$
$$+ [\sigma_n^n+(1-\sigma_n)^n]\sum_{j=1}^{\alpha} Y_{nj}L_{n-j}$$
$$+ \sum_{i=1}^{\alpha-1} W \sum_{j=0}^{i} Y_{ij}L_{i-j} + \sum_{i=\alpha}^{n_o-1} W \sum_{j=0}^{\alpha} Y_{ij}L_{i-j}$$
$$+ \sum_{i=n_o}^{n_o+\alpha} W[\sum_{j=0}^{i-n_o} Y_{ij}L_{i-j} + \sum_{j=i-n_o+1}^{\alpha} Y_{ij}L_{i-j}]$$
$$+ \sum_{i=n_o+\alpha+1}^{n-1} W \sum_{j=0}^{\alpha} Y_{ij}L_{i-j} \Big\} \qquad (4.9)$$

By substituting (2.26) into (4.9), we obtain

$$\bar{L}_n \le [1/\{1-Y_{n0}(1-\sigma_n)^n - Y_{n0}\sigma_n^n\}] \Big\{ 2$$

$$+ (\sigma_n^n + (1-\sigma_n)^n] \sum_{j=1}^{\alpha} Y_{nj}[\beta_{un_o}(n-j)-C]$$

$$+ \sum_{i=1}^{\alpha-1} W \sum_{j=0}^{i} Y_{ij}L_{i-j} + \sum_{i=\alpha}^{n_o-1} W \sum_{j=0}^{\alpha} Y_{ij}L_{i-j}$$

$$+ \sum_{i=n_o}^{n_o+\alpha} W[\sum_{j=0}^{i-n_o} Y_{ij}\{\beta_{un_o}(i-j)-C\} + \sum_{j=i-n_o+1}^{\alpha} Y_{ij}L_{i-j}]$$

$$+ \sum_{i=n_o+\alpha+1}^{n-1} W \sum_{j=0}^{\alpha} Y_{ij}[\beta_{un_o}(i-j)-C] \Big\} \qquad (4.10)$$

After some algebraic manipulation, we obtain the upper bound of $\bar{L}_n$ for this case,

$$\beta_{un_o} = \sup_{n \ge n_o+\alpha+1} [N / D] \qquad (4.11)$$

where $N = 1 + \sigma_n^n + (1-\sigma_n)^n + \sum_{i=1}^{n_o-1} W \sum_{j=0}^{\min(\alpha,i)} Y_{ij}\ (L_{i-j} + C\ )$

$$+ \sum_{i=n_o}^{n_o+\alpha} W \sum_{j=i-n_o+1}^{\alpha} Y_{ij}(L_{i-j}+C)$$

$$D = n[\sigma_n(1-\sigma_n/\alpha)^{n-1} + (1-\sigma_n)\{1-(1-\sigma_n)/\alpha\}^{n-1}]$$

$$+ \sum_{i=1}^{n_o-1} W \sum_{j=0}^{\min(\alpha,i)} Y_{ij}(i-j)$$

Obviously lower bound of $\bar{L}_n$ should be

$$\beta_{ln_o} = \inf_{n \ge n_o+\alpha+1} [N / D] \qquad (4.12)$$

Now we need to find the value of C for this protocol. For a sufficiently large "n", the "n" colliding users will be divided approximately evenly into 2α groups each with approximately $n/2\alpha$ users, i.e.,

$$\bar{L}_n \cong 2 + 2\alpha\ L_{n/2\alpha}$$

- claim : $\bar{L}_n \cong \beta n - 2/(2\alpha-1)$
- check : $2 + 2\alpha L_{n/2\alpha} = 2 + 2\alpha\,[\beta n/2\alpha - 2/(2\alpha-1)]$

$$= \beta n - 2/(2\alpha-1)$$

Therefore, a better selection of C for this protocol is

$C = 2/(2\alpha-1)$.

V. NUMERICAL RESULTS AND DISCUSSION

We tackled the analysis of the proposed multichannel collision resolution algorithm in three protocols as described in the previous chapter. We will present the corresponding numerical results, compare with computer simulation and discuss them accordingly in this chapter.

There are quite a number of high level programming languages available today. Some programming languages are specifically developed for simulations of complex organizations and systems, and others are designed for computer assisted instruction.

FORTRAN is one of the most widely used languages for expressing mathematical relationships. It has excellent mathematical capability and is, therefore, fully equipped and supported by our computer center. In this thesis, simulation is required to verify the results of theoretical analysis. Therefore, FORTRAN is quite suitable for this purpose.

A. RESULTS FOR M-ARY SPLITTING

Through theoretical analysis of m-ary splitting, we have several expressions for $\overline{L}_n$, β_{un_o}, etc. To compute $\overline{L}_n$ for high enough value of n, some numerical analysis techniques were needed. We obtained Y_{ij} corresponding to the number of available channel, as called α . In this thesis, α was selected the value of 2, 3, 5 and 10. Numerical values of Y_{ij}, corresponding to the above value of α , are provided in Appendix B. Some of them are small enough to be neglected, so those are represented as 0, but the actual value zero of Y_{ij} is deleted from this table.

The simulation program, written in FORTRAN, is provided in Appendix A. For dividing groups and selecting channels, pseudorandom number generator routine is used with different seed number at each time. $\bar{L}_n$ is simulated in this program.

Numerical values of $\bar{L}_n$ are compared with the simulation results. The corresponding comparisons are given in Figure 5.1, 5.2, 5.3 and 5.4 with each different number of available channels. As shown in these figures, correctness of theoretical analysis is certified, because two results are quite close to each other. Three values of "m" are considered, i.e., 2, 3 and 5. For each case of α , binary splitting shows the best performance and ternary splitting is next and so on. Also, it can be seen from these graphs that the performance improves as the number of available channels, α is increased.

Table I contains the analytical results for the linear bounding of $\bar{L}_n$. As shown in this table, if we select a much smaller value of n_o compare to n (number of colliding users), the lower bound of $\bar{L}_n$, β_{1n_o} approaches to zero. To obtain the reasonable linear bounding of $\bar{L}_n$, the differences between those two values, n_o and n, should be decreased.

The analytical results for throughput are contained in Table II. In those papers mentioned in Chapter I, throughput should be less than 1.0, because only single slotted channel was used in those algorithms. Since the multichannel random accessing problem is considered in this study, the results obtained here cannot be compared with the results in those papers. As shown in Table II, throughput is gradually increased as the number of channels are increased. The best performance was obtained with parameters of α = 10 and m = 2. Even with the same parameters, different throughputs were obtained in accordance with the value of n_o.

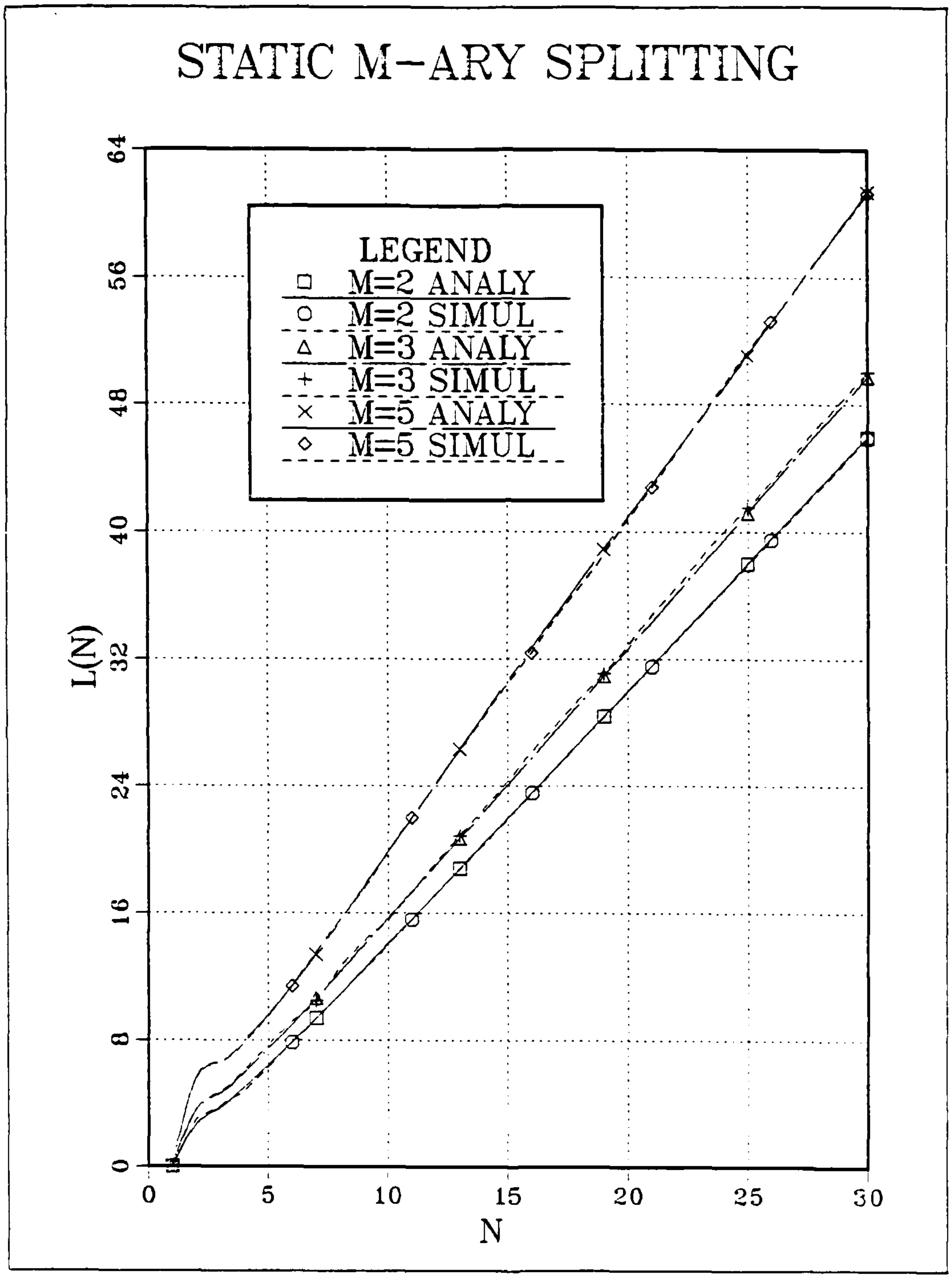


Figure 5.1 $\overline{L}_n$ for M-ary Splitting (α = 2)

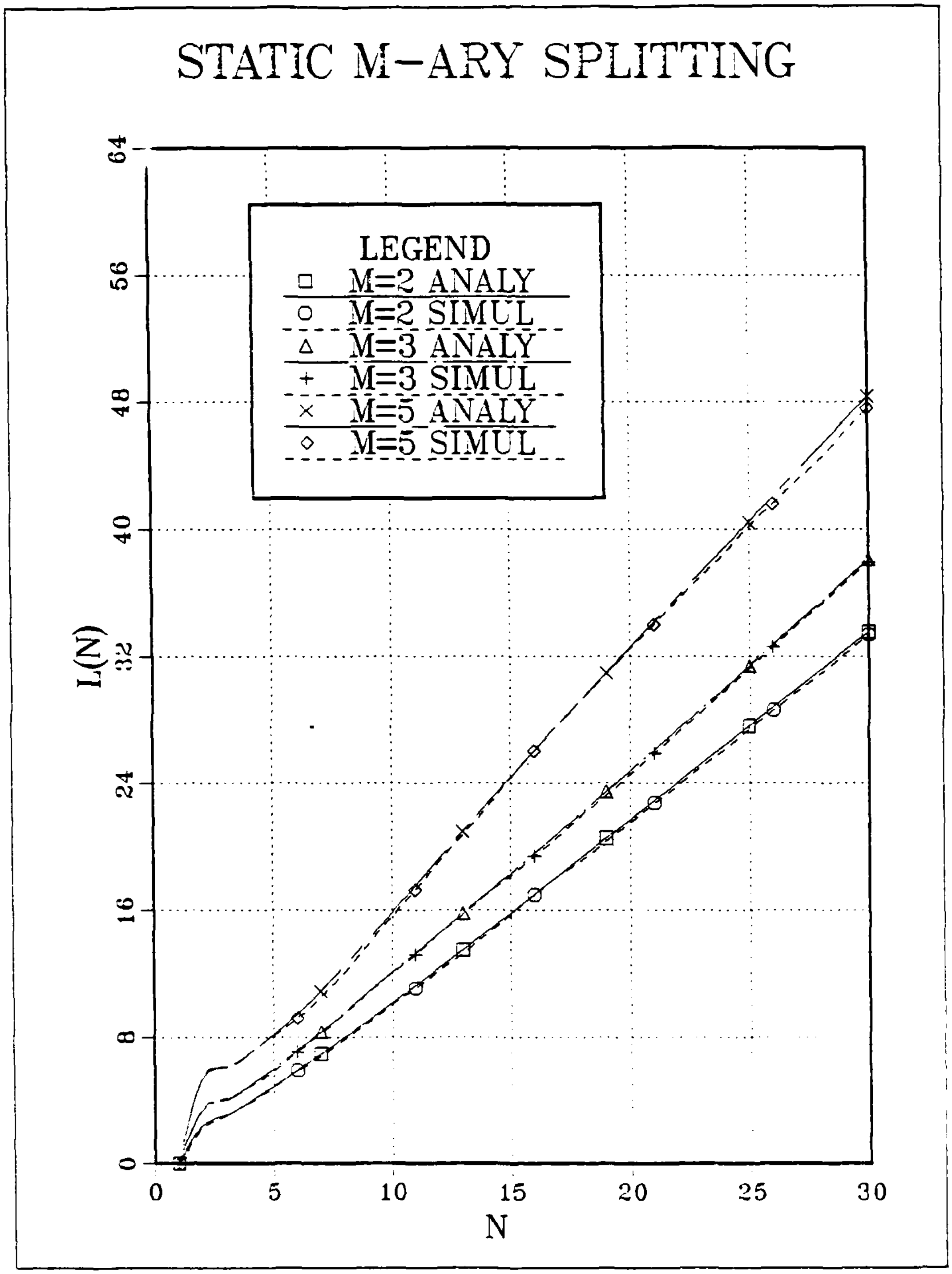


Figure 5.2 $\overline{L}_n$ for M-ary Splitting (α = 3)

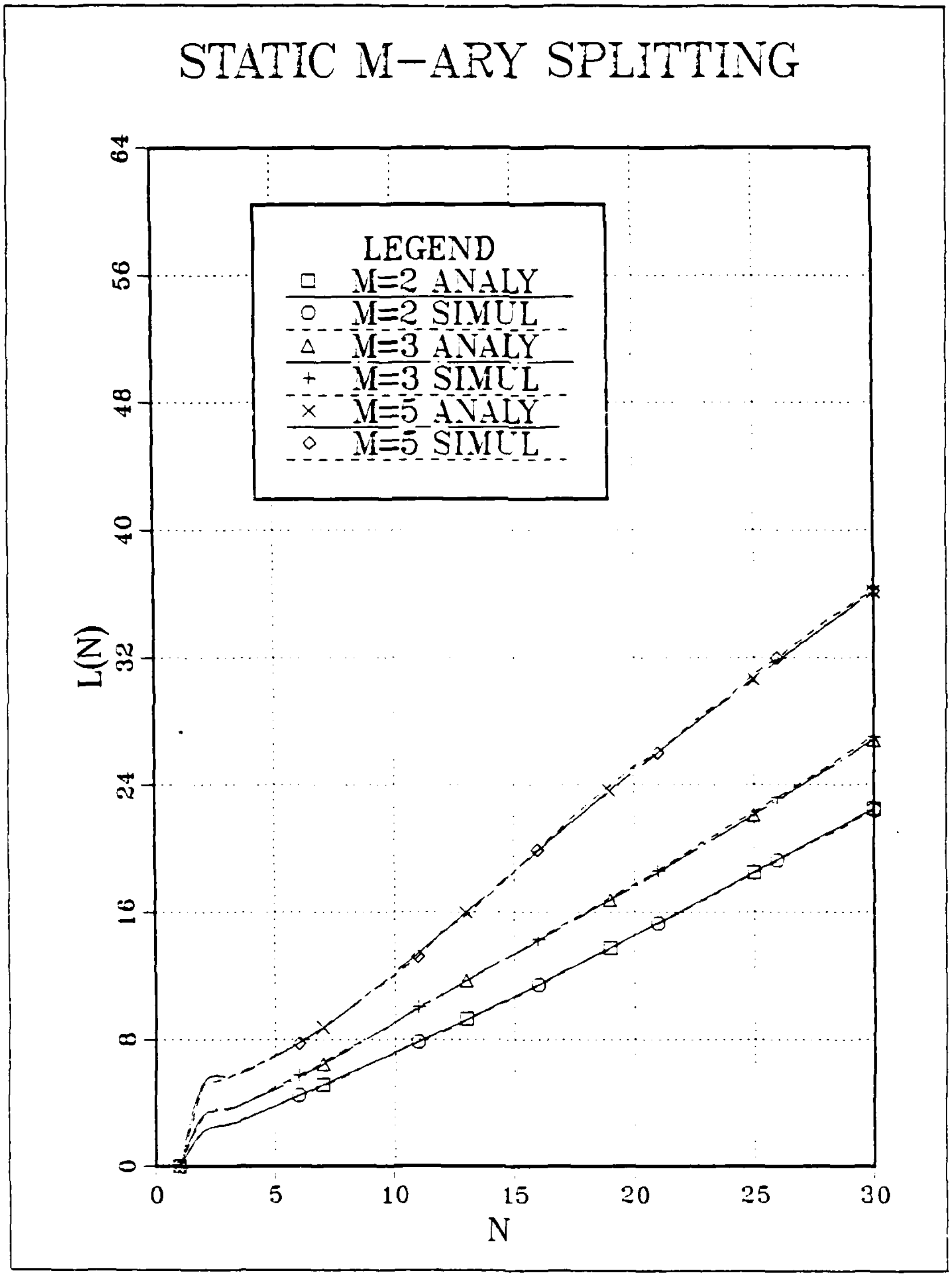


Figure 5.3 $\overline{L}_n$ for M-ary Splitting (α = 5)

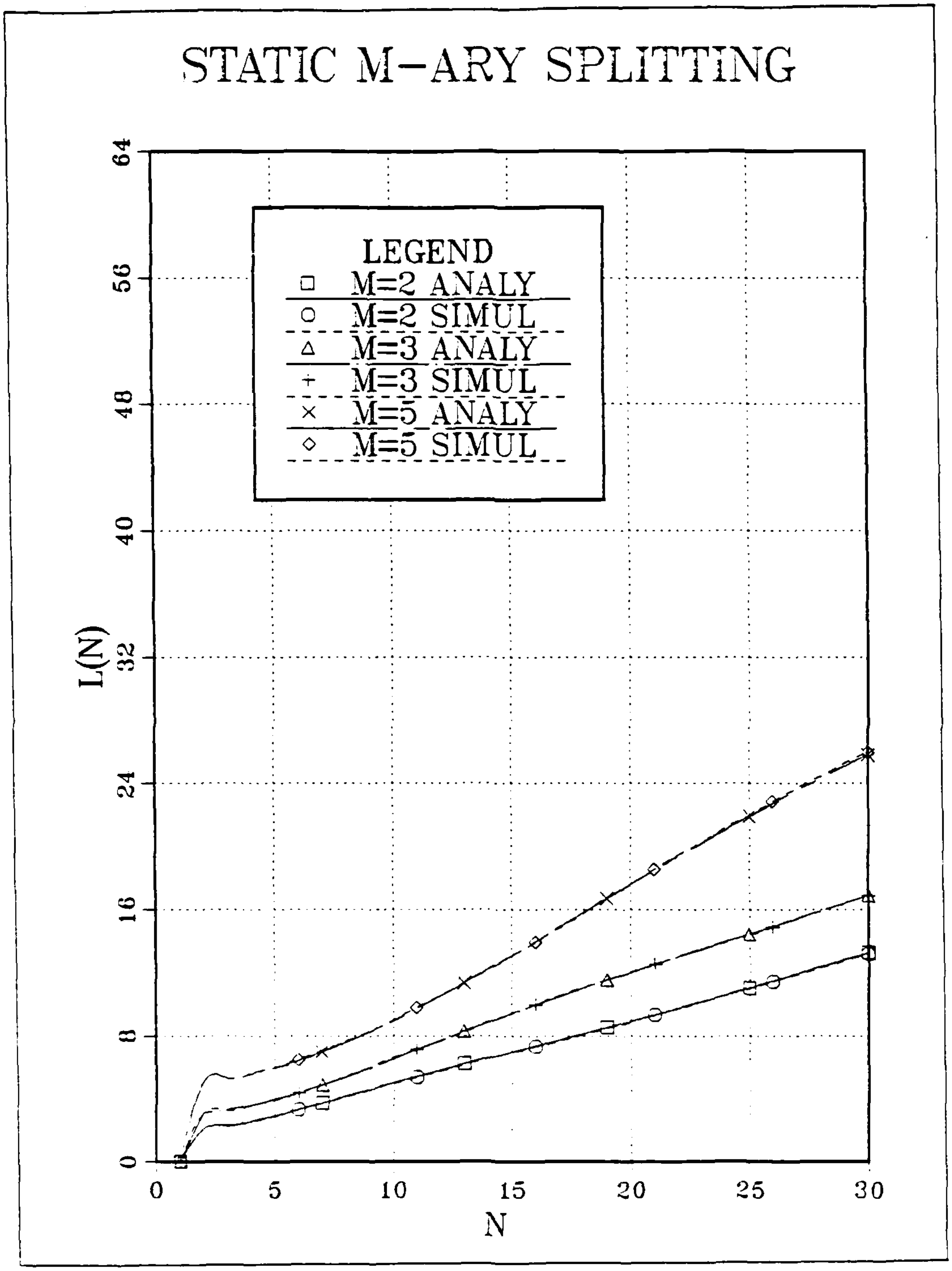


Figure 5.4 $\overline{L}_n$ for M-ary Splitting (α = 10)

TABLE I

LINEAR BOUNDING OF $\overline{L}_N$ FOR M-ARY SPLITTING

α	M	N_o	N	β_{uno}	β_{lno}
2	2	10	150	1.6027	0.0
2	2	21	110	1.59967	1.575186
2	2	21	100	1.59967	1.59656
2	2	51	150	1.5996387	1.5996025
2	3	10	150	1.7144	0.00015
2	3	15	150	1.709172	0.815373
2	3	21	110	1.709029	1.706522
2	3	21	150	1.709029	1.706522
2	5	10	150	2.121764	1.485665
2	5	15	150	2.121758	2.0847387
2	5	21	150	2.1164	2.0808
3	2	10	150	1.2141	0.00
3	2	21	150	1.185685	0.00
3	2	21	110	1.185685	0.00
3	2	31	150	1.18529	0.00874
3	2	21	100	1.185685	0.018
3	2	21	60	1.185685	1.18056
3	2	51	150	1.18528	1.18523
3	3	10	150	1.3564	0.00
3	3	21	150	1.3198	0.3150
3	3	21	110	1.3198	1.2928
3	3	51	150	1.3190	1.3145
3	5	10	150	1.7077	0.028
3	5	15	150	1.7069	1.3924
3	5	21	150	1.6954	1.6486
5	2	10	150	0.9172	0.0
5	2	10	30	0.9172	0.4112
5	2	21	50	0.8250	0.7794
5	2	41	150	0.815959	0.0134
5	2	51	150	0.815897	0.809036

TABLE I

LINEAR BOUNDING OF $\overline{L}_n$ FOR M-ARY SPLITTING(CONT'D.)

5	3	10	150	1.0572	0.0
5	3	21	150	0.95286	0.0003
5	3	31	150	0.95056	0.75214
5	3	41	150	0.95056	0.94339
5	3	51	150	0.949854	0.942393
5	5	10	150	1.341835	0.001398
5	5	21	150	1.29949	1.1474
5	5	31	150	1.276177	1.209858
5	5	41	150	1.2758	1.209859
5	5	51	150	1.27583	1.21490
10	2	10	150	0.70137	0.0
10	2	21	150	0.54	0.0
10	2	31	150	0.5071	0.0
10	2	41	150	0.4934	0.0
10	2	51	150	0.4886	0.022
10	2	61	150	0.488	0.4578
10	3	10	150	0.8026	0.0
10	3	21	150	0.6684	0.0
10	3	31	150	0.6104	0.032
10	3	41	150	0.6035	0.5267
10	3	51	150	0.6035	0.5890
10	5	11	150	1.0212	0.00105
10	5	21	150	0.9399	0.4404
10	5	31	150	0.8975	0.7782
10	5	41	150	0.8587	0.7782
10	5	51	150	0.85836	0.77818

Note : α = number of available channels

M = value of "m" in m-ary splitting

N = number of colliding packets

β_{un_o} = upper bound of $\overline{L}_n$

β_{1n_o} = lower bound of $\overline{L}_n$

TABLE II

RESULTS OF THROUGHPUT FOR M-ARY SPLITTING

α	M	N	β_{un_o}	λ_{max}
2	2	10	1.6027	0.75965567
2	2	21	1.59967	0.75965568
2	2	51	1.5996387	0.75965568
2	3	10	1.7144	0.65249
2	3	15	1.709972	0.65249
2	5	10	2.121764	0.50916
3	2	10	1.2141	0.99682
3	2	21	1.185685	0.996823
3	3	10	1.3564	0.828139
3	3	21	1.3198	0.828130
3	5	10	1.7077	0.627832
3	5	15	1.7069	0.6278323
5	2	10	0.9172	1.39498
5	3	10	1.0572	1.1289654
5	5	10	1.3418	0.843187
10	2	10	0.70137	2.3319745
10	3	10	0.8026	1.8328123
10	5	10	1.012	1.3495743
10	5	21	0.9399	1.3097
10	5	41	0.8587	1.3096

Note; λ_{max} = maximal achievable throughput

B. RESULTS OF DYNAMIC M-ARY SPLITTING

The advantage of dynamic m-ary splitting is to select the "m" in m-ary splitting as a function of "n". In other words, static m-ary splitting uses fixed m, while dynamic m-ary splitting uses m(n) to minimize $\bar{L}_n$.

Figure 5.5 contains the results of $\bar{L}_n$ which are obtained from theoretical analysis for this protocol. Through this graph, it can also be seen that the performance is increased by increasing the number of channel. As seen before, in the static m-ary splitting, binary splitting case shows the best performance. To compare with binary splitting and dynamic m-ary splitting, both data were plotted in same graph as shown in Figure 5.6. In Figure 5.6, dynamic m-ary splitting outperforms static binary splitting.

Results of linear bounding of $\bar{L}_n$ for this protocol are given in Table III. Also in throughput analysis, dynamic m-ary splitting performs better than static m-ary splitting. These results are shown in Table IV.

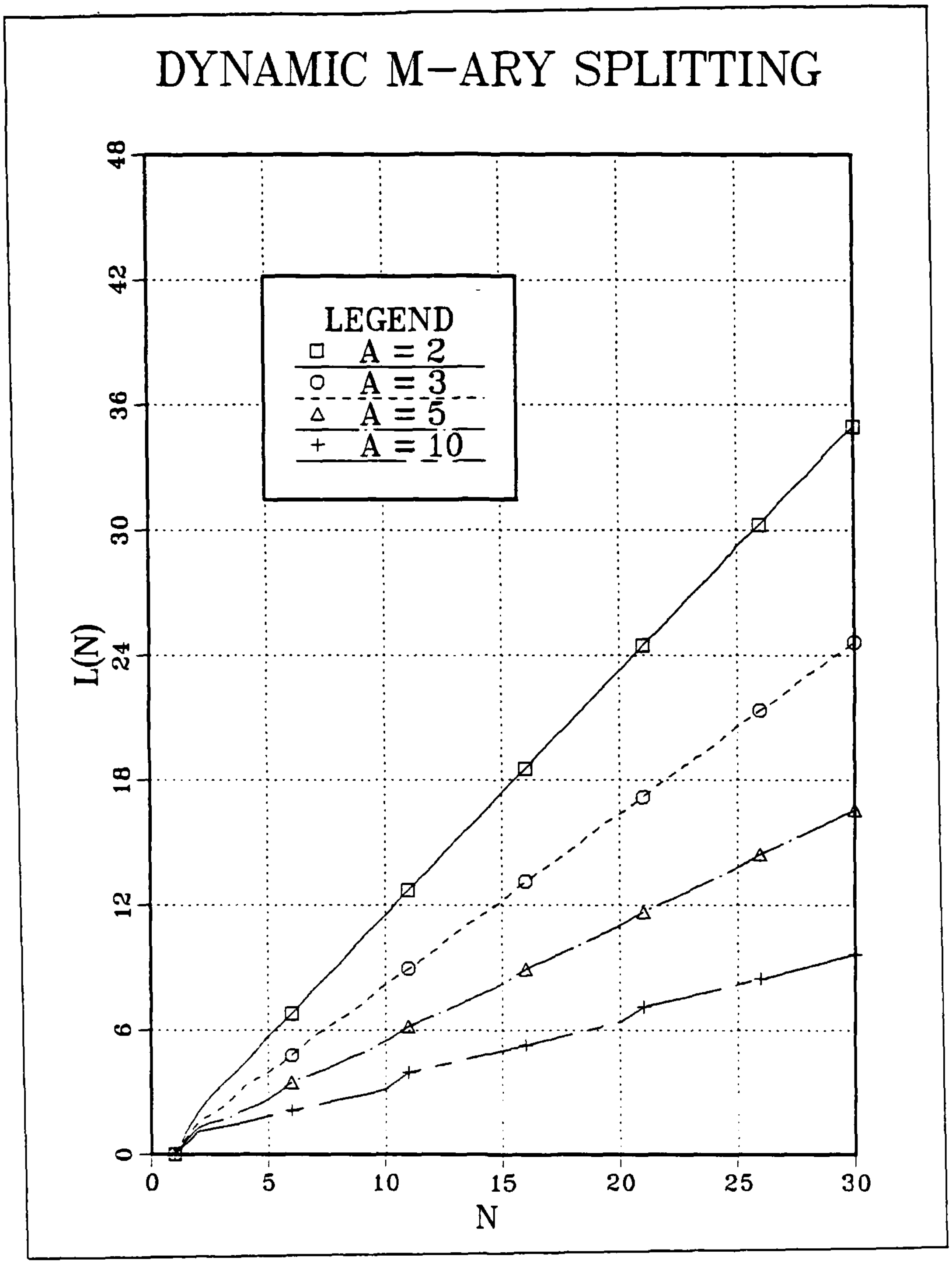


Figure 5.5 Results of $\overline{L}_n$ for Dynamic M-ary Splitting

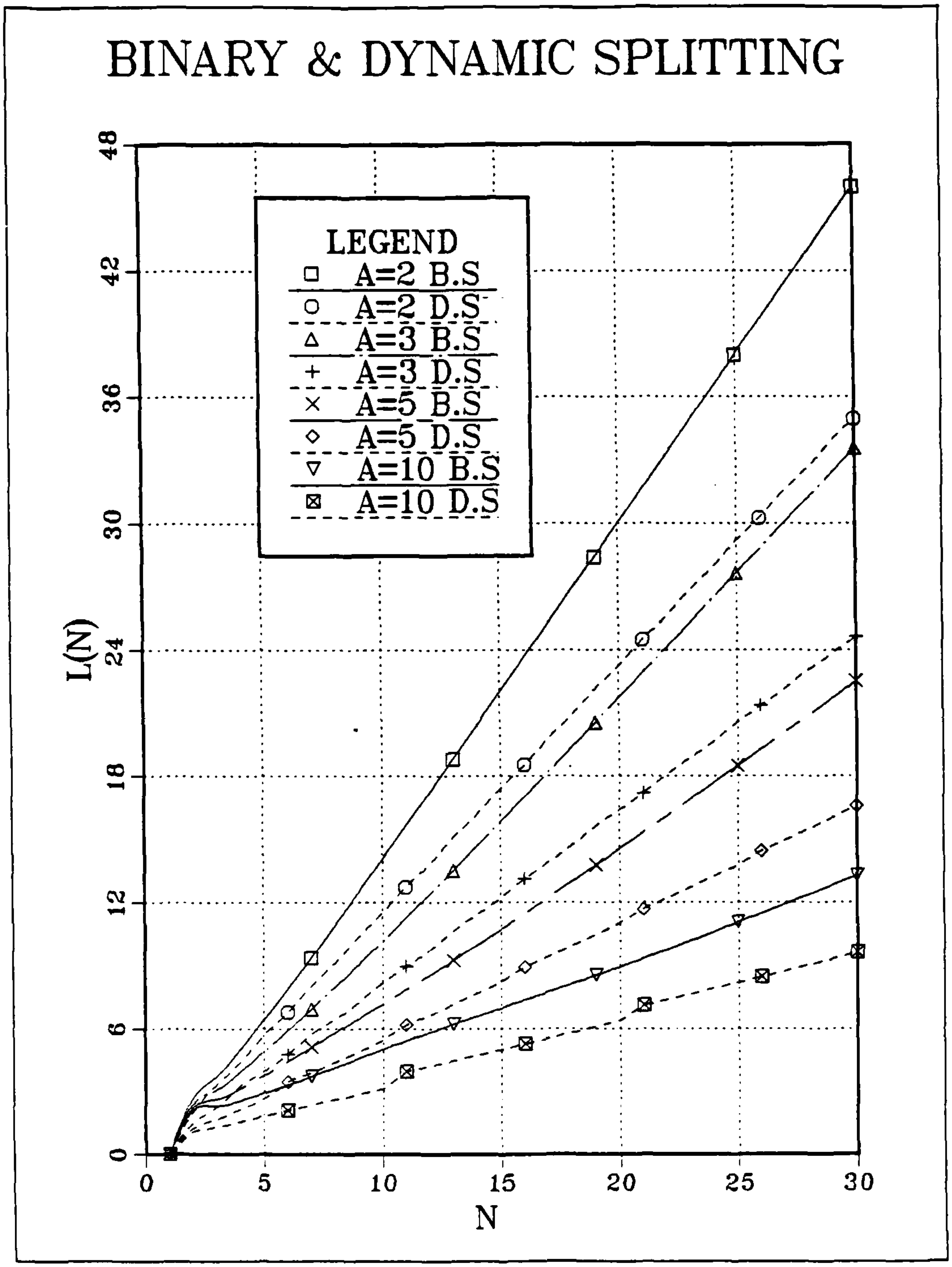


Figure 5.6 Comparison of Binary and Dynamic Splitting

TABLE III

LINEAR BOUNDING OF $\overline{L}_N$ FOR DYNAMIC M-ARY SPLITTING

α	N_o	N	β_{un_o}	β_{ln_o}
2	10	150	1.272472	1.178883
2	21	150	1.2117635	1.178896
2	31	150	1.202447	1.178896
3	10	150	0.951344	0.831984
3	21	150	0.873781	0.832144
3	31	150	0.858532	0.832144
5	10	150	0.745224	0.559788
5	21	150	0.615479	0.562131
5	31	150	0.592985	0.56213176
10	11	150	0.54172959	0.3127287
10	21	150	0.410154	0.330307
10	31	150	0.375021	0.330331
10	41	150	0.359558	0.330331

TABLE IV

RESULTS OF THROUGHPUT FOR DYNAMIC M-ARY SPLITTING

α	N	β_{un_0}	λ_{max}
2	10	1.272472	0.855374
2	21	1.217635	0.855386
3	10	0.951344	1.2111
3	21	0.873781	1.2114
5	10	0.745224	1.7848
5	21	0.615479	1.7998
5	31	0.592985	1.7998
10	11	0.54172959	2.9477998
10	21	0.410154	3.0952285
10	31	0.375021	3.097495
10	41	0.359558	3.0974956

C. RESULTS OF BINARY SPLITTING WITH ENERGY DETECTOR

In accordance with the availability of an infinite energy detector, if collision occurs, the total number of users involved in the collision are known exactly. With this advantage, the optimal σ_n, which minimizes $\bar{L}_n$, were sought for each case of α and n. For most of the cases, σ_n is very close to 0.5. If we use expression in (4.2), which is expressed in recursive form, with $\sigma_n = 0.5$, the results of $\bar{L}_n$ are exactly the same as in static binary splitting. With this result in mind, simulation is not needed for this protocol. The results of $\bar{L}_n$ for binary splitting with energy detector are given in Figure 5.7.

If we use modified binary splitting with energy detector protocol, as expressed in (4.3), value of $\bar{L}_n$ can be reduced remarkably. Figure 5.8 shows these results. However, this modified protocol is very sensitive to channel errors, as in MCCRA, because the decision of retransmission depends on the feedback information strongly.

Analytic results of $\bar{L}_n^d$ are plotted in Figure 5.9, 5.10, 5.11 and 5.12 for several values of α . $\bar{L}_n^d$, the actual average number of slots required in successfully transmitting a total of n packets including the initial slot, is commonly used for all protocols considered in this thesis.

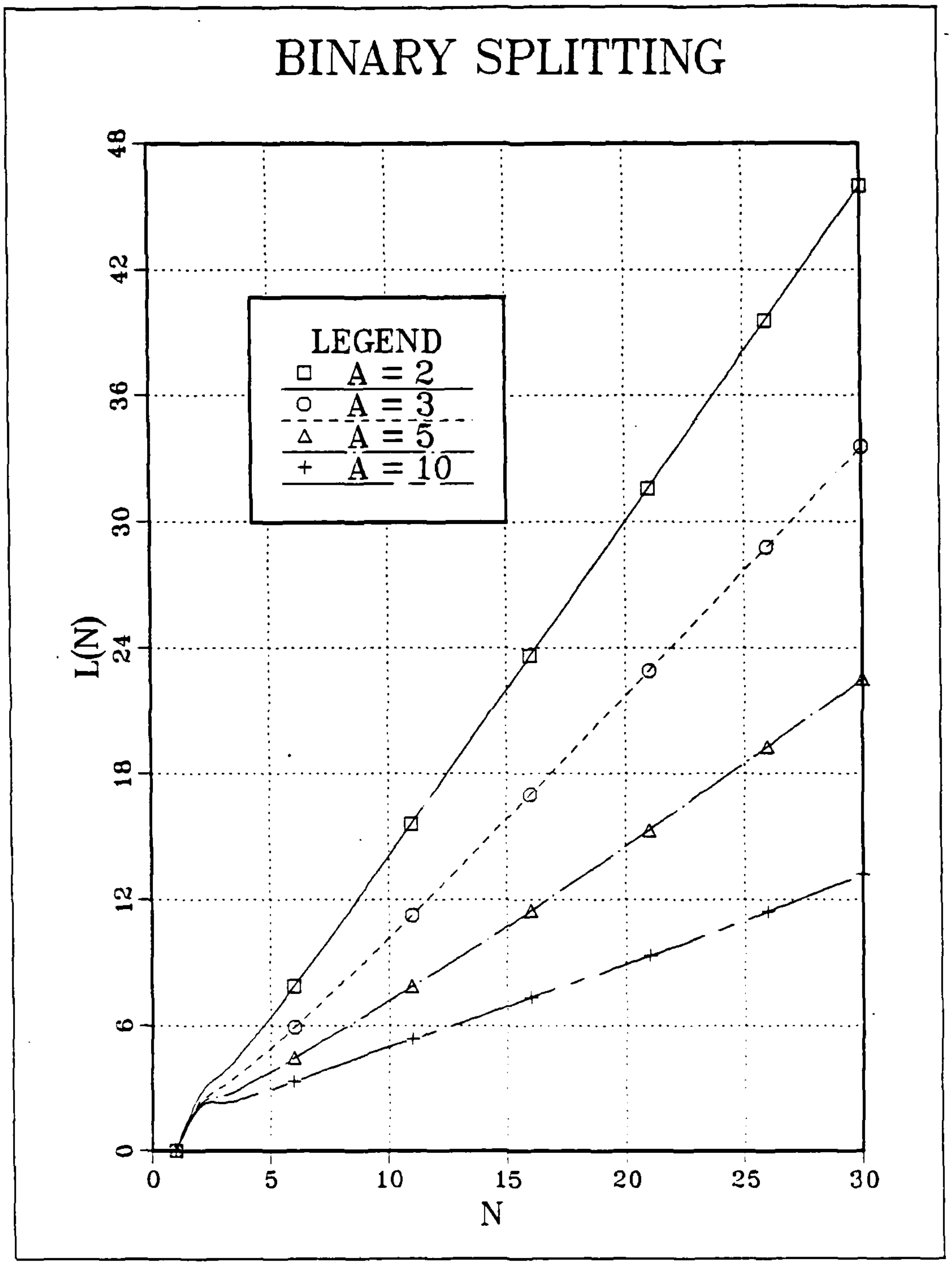


Figure 5.7 $\overline{L}_n$ for Binary Splitting with Energy Detector

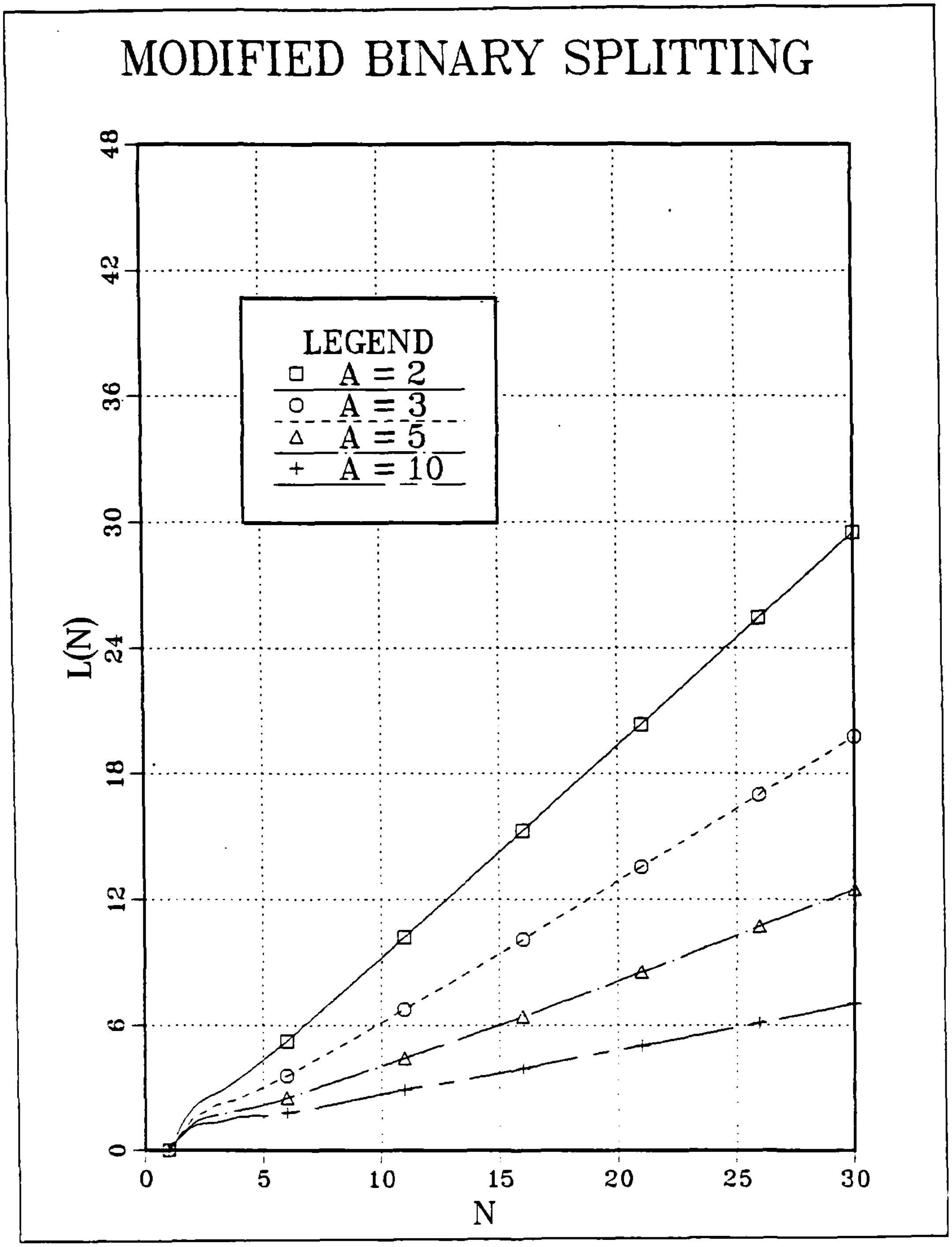


Figure 5.8 $\overline{L}_n$ for Modified Binary Splitting with Energy Detector

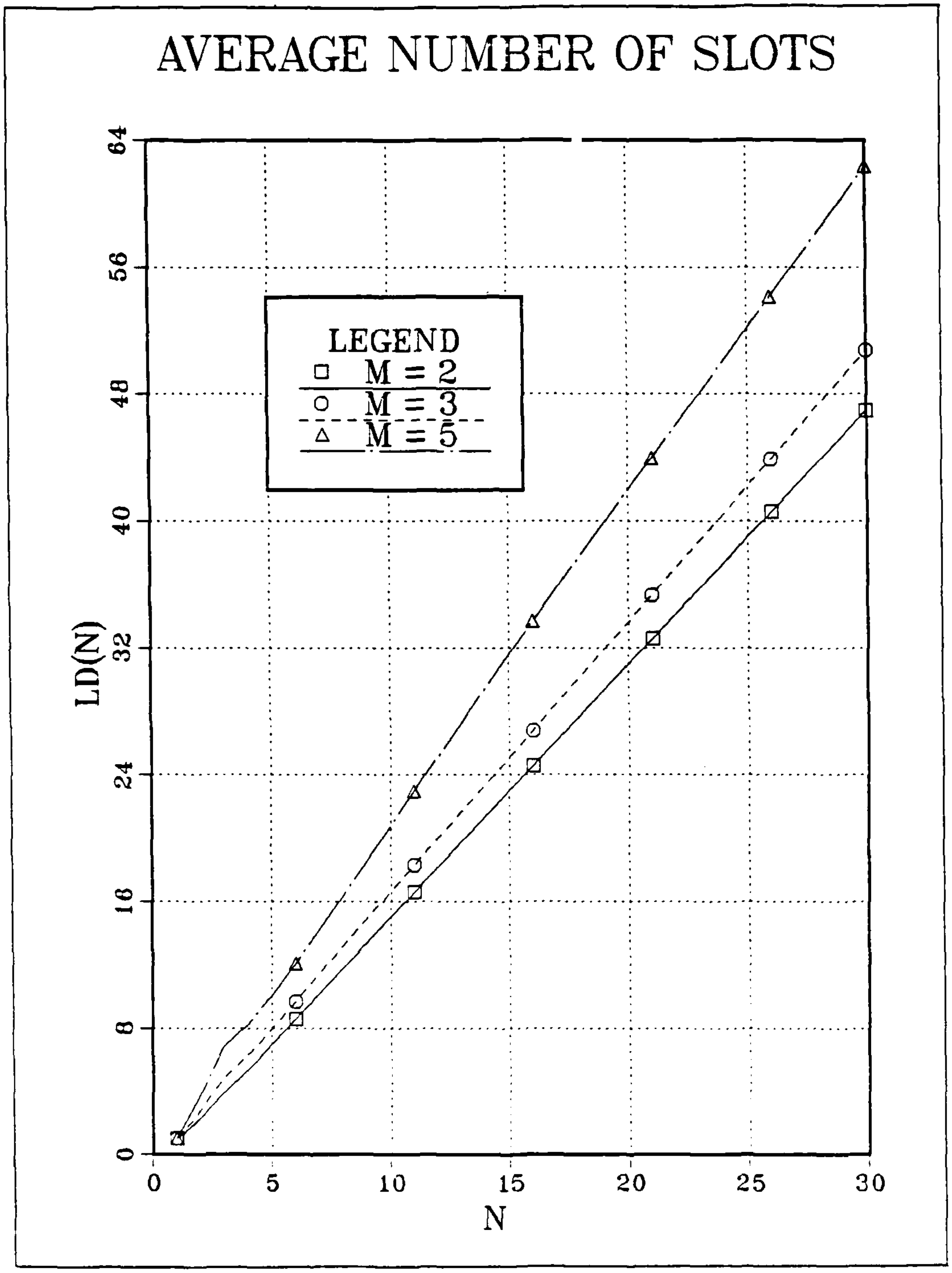


Figure 5.9 $\overline{L}_n^d$ (a = 2)

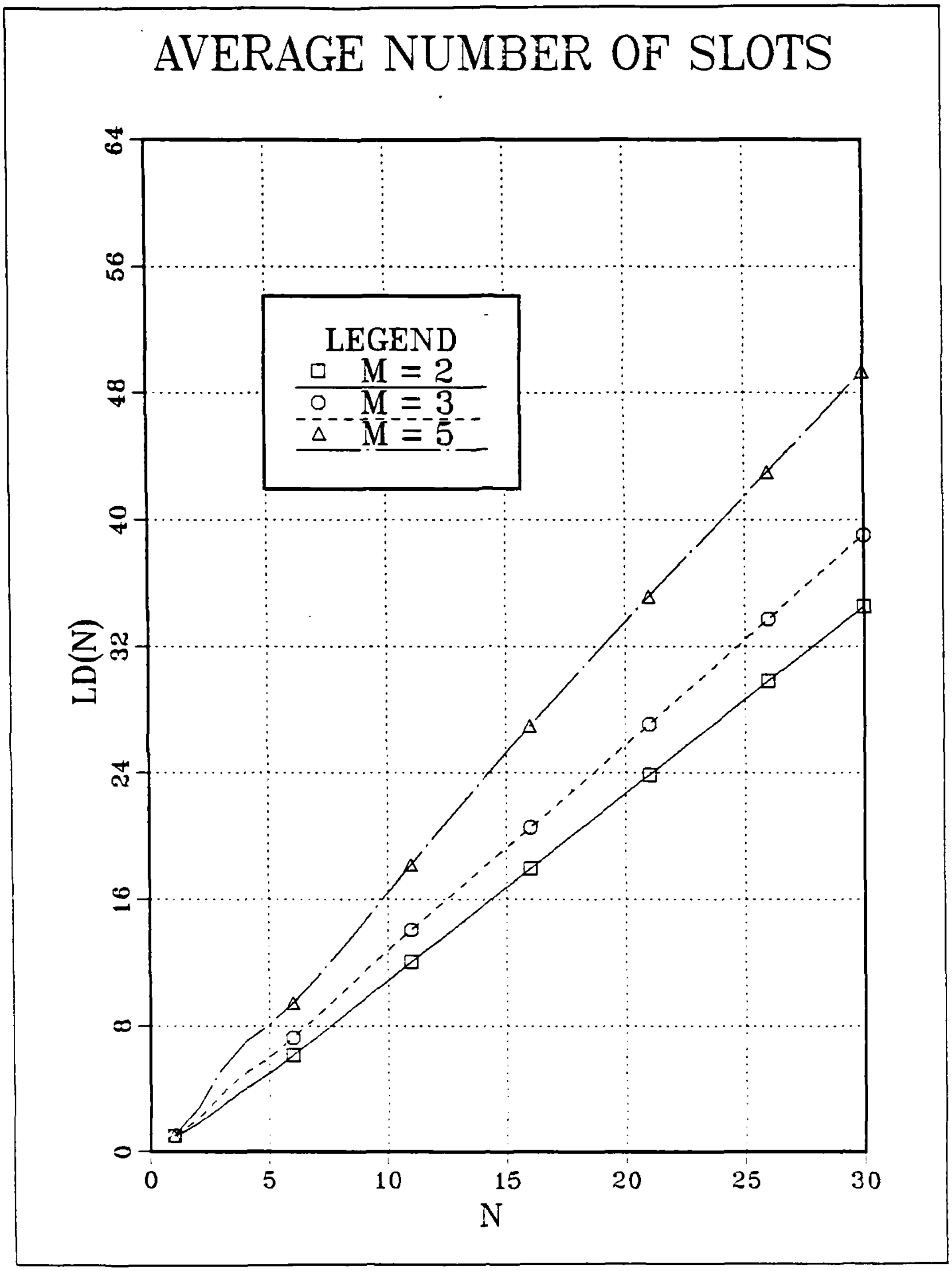


Figure 5.10 $\bar{L}_n^d$ (α = 3)

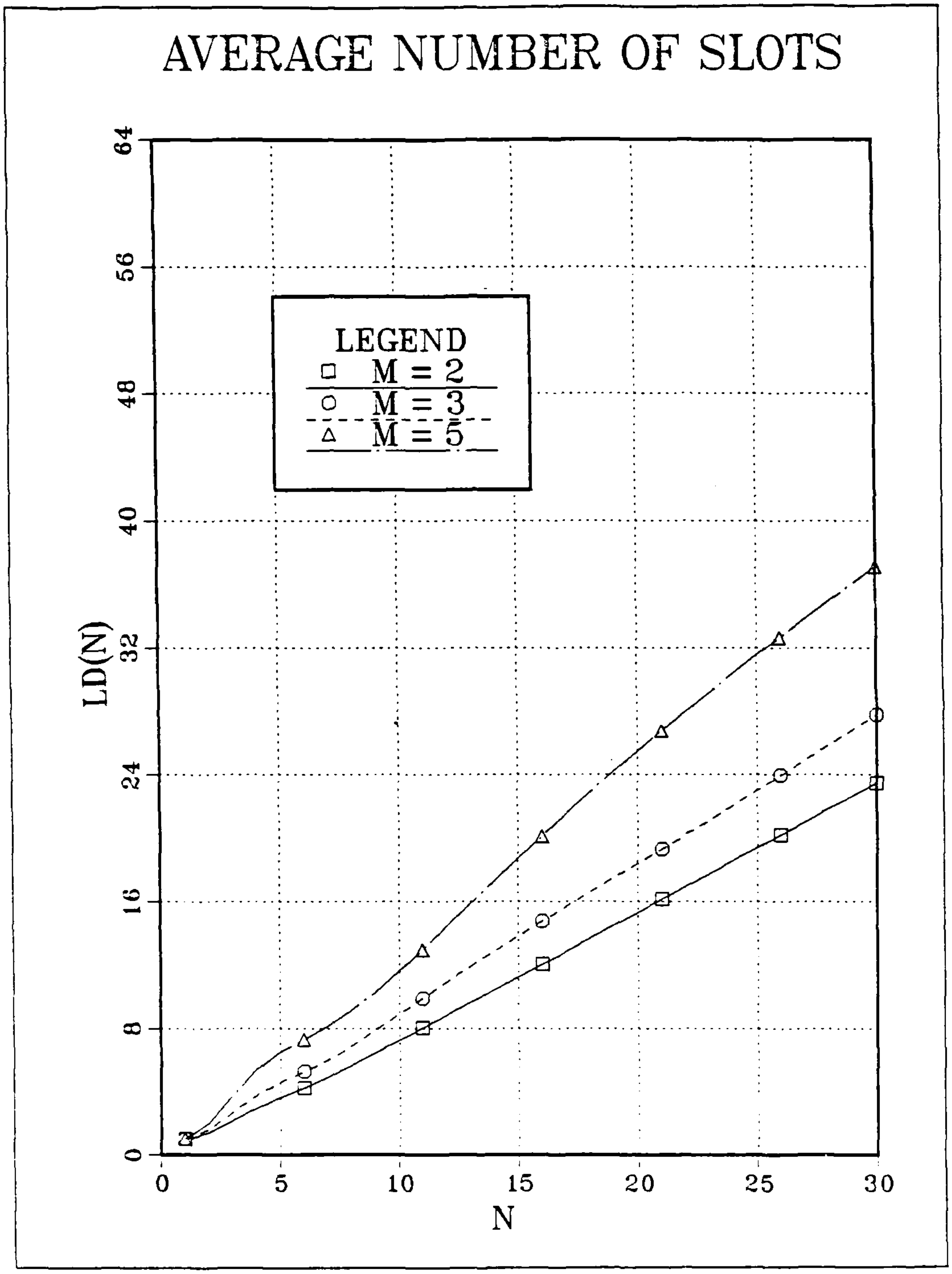


Figure 5.11 $\overline{L}_n^d$ (α = 5)

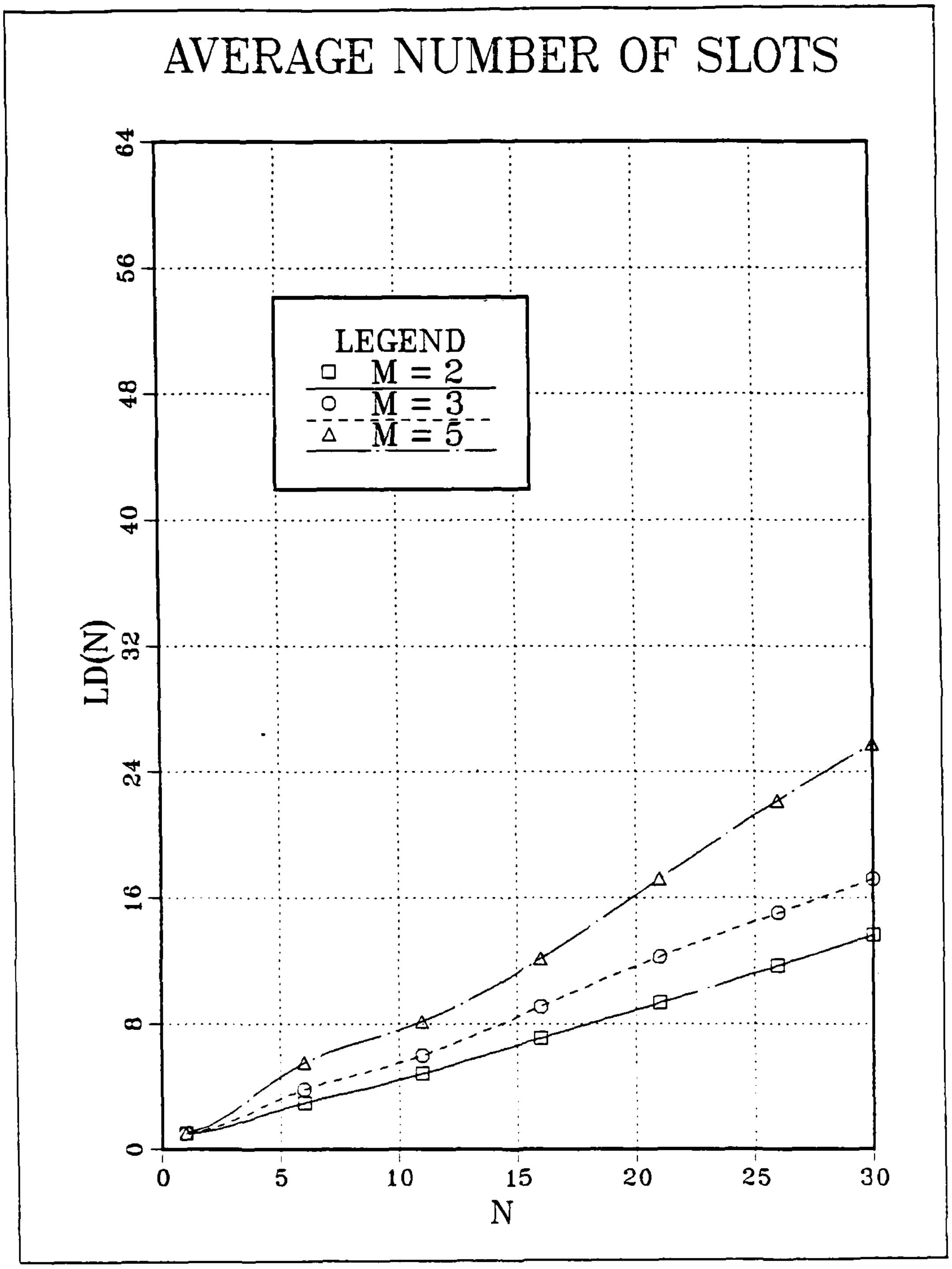


Figure 5.12 $\overline{L}_n^d$ (α = 10)

VI. CONCLUSION

This thesis has presented a few collision resolution algorithms for a multichannel random accessing communications system. Since some multichanneling can be a feasible configuration, it's worth the effort to observe the application of CRA to a multichannel system. It has been noted that there are a few ways to handle the multichannel CRA. In this thesis we focus on one class of methods which resolve collisions in all channels jointly.

We consider three different collision resolution algorithms. These algorithms have been analyzed to obtain the desired system performance by computing $\overline{L}_n$, linear bounding of $\overline{L}_n$ and maximal achievable throughput. Through the theoretical analysis and computer simulations, we have observed that binary splitting shows the best performance in static m-ary splitting. However, in applying infinite energy detectors to m-ary splitting, we are able to select the proper value of m, as a function of α and n, which minimizes the $\overline{L}_n$ and maximizes the throughput for each case. Thus, we have algorithm better than the best static m-ary splitting. This better algorithm is dynamic m-ary splitting.

Since it was proven that the best static m-ary splitting is binary splitting, the binary splitting with energy detector was also considered and evaluated. The unique feature of this algorithm is the optimal probability σ_n that a user will join the first group when the collision involves "n" users occurs. Modified binary splitting with energy detector was also analyzed by adopting Massey's idea. We observed that this modified algorithm outperforms the dynamic m-ary splitting except in the system stability. If we consider only the ideal channel, certainly this modified

algorithm will be the best one. In practice, however, it is impossible to expect an ideal channel.

When considering channel noise or channel errors, (as far as we are concerned in the multichannel collision resolution algorithms), we conclude that dynamic m-ary splitting will maintain minimum $\bar{L}_n$ and maximum achievable throughput. We note again that the throughputs computed here cannot be compared with those in several previous papers [Refs. 6, 7, 8, 9, 10, 12], because the multichannel system is much different from the single channel system. These algorithms were shown to be stable - to have maximum throughput within a certain arrival rate.

As mentioned earlier, the network performance can be evaluated by three properties - throughput, delay and stability. The maximum achievable throughput and the system stability were found, but not for average delay, in this thesis. Therefore, these proposed algorithms can be further studied for average delay in the future. However, the results developed here are applicable to a general system in which a central facility is accessible by a number of independent users. If, for example, the number of users that the facility (multichannel system) can handle simultaneously is less than the maximum number of users that can place requests upon it, then collisions will arise that may be resolved with the techniques developed here.

APPENDIX A

SIMULATION PROGRAM FOR EVALUATING M-ARY SPLITTING ALGORITHM

```
''
''DEFINITION OF VARIABLES   ************************

  A    = NUMBER OF AVAIABLE CHANNELS
  N    = NUMBER OF COLLIDING USERS
  M    = NUMBER OF SPLITTING
  LN   = TOTAL LENGTH OF SLOTS REQUIRED TO
         RESOLVE COLLISION
  L(N)= AVERAGE LENGTH OF SLOTS
''T    = NUMBER OF COMPUTING TIMES

''DECLARATION OF VARIABLES   **********************

           INTEGER N,I,T,A,T1,T2
           INTEGER T3,T4,T5,TC,M
''         REAL L(0:100),C,LN,TD,SC

''SELECT THE NUMBER OF CHANNEL AND SPLITTING   ****

         A = 2, 3, 5 or 10
         M = 2, 3, or 5
         T = 2000 OR MORE
         L(0) = 0.
         L(1) = 1.
         TD = FLOAT(T)
         DO 31 N = 2,30
          LN = 0.
          DO 41 I = 1,T
           C = 0.
           GO TO 33
'32        C = C + 1.

 33        IF(M.EQ.2) THEN
              CALL TRACH2(A,N,I,C,T1,T2)
              GO TO 34
           ELSE IF(M.EQ.3) THEN
              CALL TRACH3(A,N,I,C,T1,T2,T3)
              GO TO 35
           ELSE
              CALL TRACH5(A,N,I,C,T1,T2,T3,T4,T5)
              GO TO 36
''         END IF

 34        IF((T1.EQ.N).OR.(T2.EQ.N)) GO TO 30
              IF(A.EQ.2) THEN
                 CALL CONSA2(A,N,I,C,T1,L,SC)
                 C = C + SC
                 CALL CONSA2(A,N,I,C,T2,L,SC)
                 C = C + SC
              ELSE IF(A.EQ.3) THEN
                 CALL CONSA3(A,N,I,C,T1,L,SC)
                 C = C + SC
                 CALL CONSA3(A,N,I,C,T2,L,SC)
                 C = C + SC
              ELSE IF(A.EQ.5) THEN
                 CALL CONSA5(A,N,I,C,T1,L,SC)
                 C = C + SC
                 CALL CONSA5(A,N,I,C,T2,L,SC)
                 C = C + SC
              ELSE
```

```
          CALL CONSA0(A,N,I,C,T1,L,SC)
          C = C + SC
          CALL CONSA0(A,N,I,C,T2,L,SC)
          C = C + SC
        END IF
        GO TO 42
''
35    IF((T1.EQ.N).OR.(T2.EQ.N).OR.(T3.EQ.N)) GO TO 30
        IF(A.EQ.2) THEN
          CALL CONSA2(A,N,I,C,T1,L,SC)
          C = C + SC
          CALL CONSA2(A,N,I,C,T2,L,SC)
          C = C + SC
          CALL CONSA2(A,N,I,C,T3,L,SC)
          C = C + SC
        ELSE IF(A.EQ.3) THEN
          CALL CONSA3(A,N,I,C,T1,L,SC)
          C = C + SC
          CALL CONSA3(A,N,I,C,T2,L,SC)
          C = C + SC
          CALL CONSA3(A,N,I,C,T3,L,SC)
          C = C + SC
        ELSE IF(A.EQ.5) THEN
          CALL CONSA5(A,N,I,C,T1,L,SC)
          C = C + SC
          CALL CONSA5(A,N,I,C,T2,L,SC)
          C = C + SC
          CALL CONSA5(A,N,I,C,T3,L,SC)
          C = C + SC
        ELSE
          CALL CONSA0(A,N,I,C,T1,L,SC)
          C = C + SC
          CALL CONSA0(A,N,I,C,T2,L,SC)
          C = C + SC
          CALL CONSA0(A,N,I,C,T3,L,SC)
          C = C + SC
        END IF
      GO TO 42
''
36    IF((T1.EQ.N).OR.(T2.EQ.N).OR.(T3.EQ.N).OR.
   #      (T4.EQ.N).OR.(T5.EQ.N)) GO TO 30
        IF(A.EQ.2) THEN
          CALL CONSA2(A,N,I,C,T1,L,SC)
          C = C + SC
          CALL CONSA2(A,N,I,C,T2,L,SC)
          C = C + SC
          CALL CONSA2(A,N,I,C,T3,L,SC)
          C = C + SC
          CALL CONSA2(A,N,I,C,T4,L,SC)
          C = C + SC
          CALL CONSA2(A,N,I,C,T5,L,SC)
          C = C + SC
        ELSE IF(A.EQ.3) THEN
          CALL CONSA3(A,N,I,C,T1,L,SC)
          C = C + SC
          CALL CONSA3(A,N,I,C,T2,L,SC)
          C = C + SC
          CALL CONSA3(A,N,I,C,T3,L,SC)
          C = C + SC
          CALL CONSA3(A,N,I,C,T4,L,SC)
          C = C + SC
          CALL CONSA3(A,N,I,C,T5,L,SC)
          C = C + SC
        ELSE IF(A.EQ.5) THEN
          CALL CONSA5(A,N,I,C,T1,L,SC)
          C = C + SC
          CALL CONSA5(A,N,I,C,T2,L,SC)
          C = C + SC
          CALL CONSA5(A,N,I,C,T3,L,SC)
          C = C + SC
```

```
            CALL CONSA5(A,N,I,C,T4,L,SC)
            C = C + SC
            CALL CONSA5(A,N,I,C,T5,L,SC)
            C = C + SC
          ELSE
            CALL CONSA0(A,N,I,C,T1,L,SC)
            C = C + SC
            CALL CONSA0(A,N,I,C,T2,L,SC)
            C = C + SC
            CALL CONSA0(A,N,I,C,T3,L,SC)
            C = C + SC
            CALL CONSA0(A,N,I,C,T4,L,SC)
            C = C + SC
            CALL CONSA0(A,N,I,C,T5,L,SC)
            C = C + SC
          END IF
        GO TO 42
''
''WHEN ALL USERS COLLIDED AGAIN, RESTART SPLITTING
 30     C = C + FLOAT(M-1)
        TC = N
        IF(A.EQ.2) THEN
           CALL CONSA2(A,N,I,C,TC,L,SC)
           IF(SC.EQ.0) GO TO 32
              C = C + SC
        ELSE IF(A.EQ.3) THEN
           CALL CONSA3(A,N,I,C,TC,L,SC)
           IF(SC.EQ.0) GO TO 32
              C = C + SC
        ELSE IF(A.EQ.5) THEN
           CALL CONSA5(A,N,I,C,TC,L,SC)
           IF(SC.EQ.0) GO TO 32
              C = C + SC
        ELSE
           CALL CONSA0(A,N,I,C,TC,L,SC)
           IF(SC.EQ.0) GO TO 32
              C = C + SC
        END IF
 42     LN = LN + C
''41    CONTINUE
''COMPUTE  L̄n  AND PRINT IT OUT   ***************

        L(N) = LN / TD
        WRITE(4,100)N,L(N)
 31    CONTINUE
 100   FORMAT(4X,I3,3X,F10.4)
        STOP
        END
''
''SUBROUTINE OF BINARY SPLITTING  *****************

        SUBROUTINE TRACH2(A,N,I,C,TC1,TC2)
        INTEGER J,TC1,TC2,N,I,A
        REAL NR(100),C
        REAL*8 DSEED
        DSEED = 182669.D0+DFLOAT(I)*12
     #      +DBLE(C)*14152+DFLOAT(A)*328
        CALL GGUBS(DSEED,N,NR)
        TC1 = 0
        TC2 = 0
        DO 201 J = 1,N
           IF(NR(J).LT.0.5) THEN
             TC1 = TC1 + 1
           ELSE
             TC2 = TC2 + 1
           END IF
 201    CONTINUE
        RETURN
```

```
,,        END
,,SUBROUTINE OF TERNARY SPLITTING ****************
          SUBROUTINE TRACH3(A,N,I,C,TC1,TC2,TC3)
          INTEGER I,J,TC1,TC2,TC3,N,A
          REAL NR(100),C
          REAL*8 DSEED
          DSEED = 224027.D0+DFLOAT(I)*108
     #            +DBLE(C)*13502+DFLOAT(A)*490
          CALL GGUBS(DSEED,N,NR)
          TC1 = 0
          TC2 = 0
          TC3 = 0
          DO 301 J = 1,N
             IF(NR(J).LT.0.3334) THEN
                TC1 = TC1 + 1
             ELSE IF(NR(J).LT.0.6667) THEN
                TC2 = TC2 + 1
             ELSE
                TC3 = TC3 + 1
             END IF
 301      CONTINUE
          RETURN
          END
,,
,,SUBROUTINE OF SPLITTING WITH FIVE BRANCHES  *******
          SUBROUTINE TRACH5(A,N,I,C,TC1,TC2,TC3,TC4,TC5)
          INTEGER I,J,TC1,TC2,TC3,TC4,TC5,N,M,A
          REAL NR(100),C
          REAL*8 DSEED
          DSEED = 198463.D0+DFLOAT(I)*327
     #            +DBLE(C)*8950+DFLOAT(A)*831
          CALL GGUBS(DSEED,N,NR)
          TC1 = 0
          TC2 = 0
          TC3 = 0
          TC4 = 0
          TC5 = 0
          DO 501 J = 1,N
             IF(NR(J).LT.0.2) THEN
                TC1 = TC1 + 1
             ELSE IF(NR(J).LT.0.4) THEN
                TC2 = TC2 + 1
             ELSE IF(NR(J).LT.0.6) THEN
                TC3 = TC3 + 1
             ELSE IF(NR(J).LT.0.8) THEN
                TC4 = TC4 + 1
             ELSE
                TC5 = TC5 + 1
             END IF
 501      CONTINUE
          RETURN
          END
,,
,,DISTRIBUTE COLLIDING USERS INTO 10 CHANNELS ********
          SUBROUTINE ASH10(A,N,I,C,TC1,TC2,TC3,TC4,TC5,
     #                   TC6,TC7,TC8,TC9,TC10)
          INTEGER I,TC1,TC2,TC3,TC4,TC5,TC6,TC7,TC8,J
          INTEGER TC9,TC10,A,N
          REAL NR(100),C
          REAL*8 DSEED
          DSEED = 123719.D0+DFLOAT(I)*119
     #            +DBLE(C)*9171+DFLOAT(A)*508
          CALL GGUBS(DSEED,N,NR)
          TC1 = 0
          TC2 = 0
          TC3 = 0
```

```
          TC4 = 0
          TC5 = 0
          TC6 = 0
          TC7 = 0
          TC8 = 0
          TC9 = 0
          TC10 = 0
          DO 601 J = 1,N
             IF(NR(J).LE.0.1) THEN
               TC1 = TC1 + 1
             ELSE IF(NR(J).LE.0.2) THEN
               TC2 = TC2 + 1
             ELSE IF(NR(J).LE.0.3) THEN
               TC3 = TC3 + 1
             ELSE IF(NR(J).LE.0.4) THEN
               TC4 = TC4 + 1
             ELSE IF(NR(J).LE.0.5) THEN
               TC5 = TC5 + 1
             ELSE IF(NR(J).LE.0.6) THEN
               TC6 = TC6 + 1
             ELSE IF(NR(J).LE.0.7) THEN
               TC7 = TC7 + 1
             ELSE IF(NR(J).LE.0.8) THEN
               TC8 = TC8 + 1
             ELSE IF(NR(J).LE.0.9) THEN
               TC9 = TC9 + 1
             ELSE
               TC10 = TC10 + 1
             END IF
601       CONTINUE
          RETURN
          END
''
  DISTRIBUTE COLLIDING USERS AND SECCEEDING USERS
''IN TWO CHANNELS      ***********************************

          SUBROUTINE CONSA2(A,N,I,C,T,L,SC)
          INTEGER N,T,CH1,CH2,I,A,COLL
          REAL C,SC,L(0:100)
          IF(T.LE.1) GO TO 433
          CALL TRACH2(A,T,I,C,CH1,CH2)
          IF((CH1.EQ.N).OR.(CH2.EQ.N)) GO TO 43
             COLL = 0
             IF(CH1.LE.1) GO TO 12
                COLL = COLL + CH1
12           IF(CH2.LE.1) GO TO 13
                COLL = COLL + CH2
                SC = 1. + L(COLL)
                GO TO 44
13           SC = 1. + L(COLL)
             GO TO 44
43        SC = 0.
          GO TO 44
433       SC = 1.
44        RETURN
          END
''
  DISTRIBUTE COLLIDING USERS AND SUCCEEDING USERS
''IN THREE CHANNELS    ***********************************

          SUBROUTINE CONSA3(A,N,I,C,T,L,SC)
          INTEGER N,A,T,I,CH1,CH2,CH3,COLL
          REAL C,SC,L(0:100)
          IF(T.LE.1) GO TO 455
          CALL TRACH3(A,T,I,C,CH1,CH2,CH3)
          IF((CH1.EQ.N).OR.(CH2.EQ.N).OR.(CH3.EQ.N))
    #        GO TO 45
          COLL = 0
          IF(CH1.LE.1) GO TO 14
             COLL = COLL + CH1
```

```
14        IF(CH2.LE.1) GO TO 15
             COLL = COLL + CH2
15        IF(CH3.LE.1) GO TO 16
             COLL = COLL + CH3
             SC = 1.+ L(COLL)
             GO TO 46
16        SC = 1.+ L(COLL)
          GO TO 46
45        SC = 0.
          GO TO 46
455       SC = 1.
46        RETURN
          END
''
  DISTRIBUTE COLLIDING USERS AND SUCCEEDING USERS
''IN FIVE CHANNELS   **********************************

          SUBROUTINE CONSA5(A,N,I,C,T,L,SC)
          INTEGER N,I,T,CH1,CH2,CH3,CH4,CH5,A,COLL
          REAL C,SC,L(0:100)
          IF(T.LE.1) GO TO 477
          CALL TRACH5(A,T,I,C,CH1,CH2,CH3,CH4,CH5)
          IF((CH1.EQ.N).OR.(CH2.EQ.N).OR.(CH3.EQ.N).OR.
     #       (CH4.EQ.N).OR.(CH5.EQ.N)) GO TO 47
          COLL = 0
          IF(CH1.LE.1) GO TO 21
             COLL = COLL + CH1
21        IF(CH2.LE.1) GO TO 22
             COLL = COLL + CH2
22        IF(CH3.LE.1) GO TO 23
             COLL = COLL + CH3
23        IF(CH4.LE.1) GO TO 24
             COLL = COLL + CH4
24        IF(CH5.LE.1) GO TO 25
             COLL = COLL + CH5
             SC = 1.+ L(COLL)
             GO TO 26
25        SC = 1.+ L(COLL)
          GO TO 26
477       SC = 1.
          GO TO 26
47        SC = 0.
26        RETURN
          END
''
  DISTRIBUTE SUCCEEDING USERS AND COLLIDING USERS
''IN TEN CHANNELS      *********************************

          SUBROUTINE CONSA0(A,N,I,C,T,L,SC)
          INTEGER N,I,T,A,CH1,CH2,CH3,CH4,CH5,CH6,CH7
          INTEGER CH8,CH9,CH10,COLL
          REAL C,SC,L(0:100)
          IF(T.LE.1) GO TO 18
          CALL ASH10(A,T,I,C,CH1,CH2,CH3,CH4,CH5,CH6,
     #                CH7,CH8,CH9,CH10)
          IF((CH1.EQ.N).OR.(CH2.EQ.N).OR.(CH3.EQ.N).OR.
     #       (CH4.EQ.N).OR.(CH5.EQ.N).OR.(CH6.EQ.N).OR.
     #       (CH7.EQ.N).OR.(CH8.EQ.N).OR.(CH9.EQ.N).OR.
     #       (CH10.EQ.N)) GO TO 19
          COLL = 0
          IF(CH1.LE.1) GO TO 1
             COLL = COLL + CH1
1         IF(CH2.LE.1) GO TO 2
             COLL = COLL + CH2
2         IF(CH3.LE.1) GO TO 3
             COLL = COLL + CH3
3         IF(CH4.LE.1) GO TO 4
             COLL = COLL + CH4
4         IF(CH5.LE.1) GO TO 5
             COLL = COLL + CH5
```

```
5        IF(CH6.LE.1) GO TO 6
           COLL = COLL + CH6
6        IF(CH7.LE.1) GO TO 7
           COLL = COLL + CH7
7        IF(CH8.LE.1) GO TO 8
           COLL = COLL + CH8
8        IF(CH9.LE.1) GO TO 9
           COLL = COLL + CH9
9        IF(CH10.LE.1) GO TO 10
           COLL = COLL + CH10
           SC = 1. + L(COLL)
           GO TO 11
10       SC = 1.+ L(COLL)
         GO TO 11
18       SC = 1.
         GO TO 11
19       SC = 0.
11       RETURN
         END
```

APPENDIX B

NUMERICAL VALUES OF Y_{IJ}

Y_{ij} denotes the probability that j users succeed when i users attempt their retransmissions in a slot which contains α channels. Using the equation (2.1), we computed Y_{ij} corresponding to each value of α (1 to 10) where n (collising users) is up to 150. Since the total number of these values is too big to attach here, this appendix contains only some of them for the value of α is 2, 3 and 5. As has been noted before, some of the values are small enough to be neglected, so they are represented as 0 in following table. On the other hand, some probabilities are very close to 1.0. Hence they are represented as 1.0. These values are valid for all protocols studied in this thesis.

TABLE V

Y_{IJ} (a = 2)

a	i	j	Y_{ij}	a	i	j	Y_{ij}
2	1	1	0.10000000D+01	2	24	0	0.99999714D+00
2	2	0	0.49999997D+00	2	24	1	0.28610229D-05
2	2	2	0.50000003D+00	2	25	0	0.99999851D+00
2	3	0	0.25000000D+00	2	25	1	0.14901161D-05
2	3	1	0.75000000D+00	2	26	0	0.99999923D+00
2	4	0	0.50000000D+00	2	26	1	0.77486038D-06
2	4	1	0.50000000D+00	2	27	0	0.99999960D+00
2	5	0	0.68750000D+00	2	27	1	0.40233135D-06
2	5	1	0.31250000D+00	2	28	0	0.99999979D+00
2	6	0	0.81250000D+00	2	28	1	0.20861626D-06
2	6	1	0.18750000D+00	2	29	0	0.99999989D+00
2	7	0	0.89062500D+00	2	29	1	0.10803342D-06
2	7	1	0.10937500D+00	2	30	0	0.99999994D+00
2	8	0	0.93750000D+00	2	30	1	0.55879354D-07
2	8	1	0.62500000D-01	2	31	0	0.99999997D+00
2	9	0	0.96484375D+00	2	31	1	0.28871000D-07
2	9	1	0.35156250D-01	2	32	0	0.99999999D+00
2	10	0	0.98046875D+00	2	32	1	0.14901161D-07
2	10	1	0.19531250D-01	2	33	0	0.99999999D+00
2	11	0	0.98925781D+00	2	33	1	0.76834112D-08
2	11	1	0.10742187D-01	2	34	0	0.10000000D+01
2	12	0	0.99414062D+00	2	34	1	0.39581209D-08
2	12	1	0.58593750D-02	2	35	0	0.10000000D+01
2	13	0	0.99682617D+00	2	35	1	0.20372681D-08
2	13	1	0.31738281D-02	2	36	0	0.10000000D+01
2	14	0	0.99829102D+00	2	36	1	0.10477379D-08
2	14	1	0.17089844D-02	2	37	0	0.10000000D+01
2	15	0	0.99908447D+00	2	37	1	0.53842086D-09
2	15	1	0.91552734D-03	2	38	0	0.10000000D+01
2	16	0	0.99951172D+00	2	38	1	0.27648639D-09
2	16	1	0.48828125D-03	2	39	0	0.10000000D+01
2	17	0	0.99974060D+00	2	39	1	0.14188117D-09
2	17	1	0.25939941D-03	2	40	0	0.10000000D+01
2	18	0	0.99986267D+00	2	40	1	0.72759576D-10
2	18	1	0.13732910D-03	2	41	0	0.10000000D+01
2	19	0	0.99992752D+00	2	41	1	0.37289283D-10
2	19	1	0.72479248D-04	2	42	0	0.10000000D+01
2	20	0	0.99996185D+00	2	42	1	0.19099389D-10
2	20	1	0.38146973D-04	2	43	0	0.10000000D+01
2	21	0	0.99997997D+00	2	43	1	0.97770680D-11
2	21	1	0.20027161D-04	2	44	0	0.10000000D+01
2	22	0	0.99998951D+00	2	44	1	0.50022209D-11
2	22	1	0.10490417D-04	2	45	0	0.10000000D+01
2	23	0	0.99999452D+00	2	45	1	0.25579538D-11
2	23	1	0.54836273D-05	2	46	0	0.10000000D+01

TABLE V

Y_{ij} (α = 2) (CONT'D.)

2	46	1	0.13073986D-11	2	71	0	0.10000000D+01
2	47	0	0.10000000D+01	2	71	1	0.60139339D-19
2	47	1	0.66791017D-12	2	72	0	0.10000000D+01
2	48	0	0.10000000D+01	2	72	1	0.30493186D-19
2	48	1	0.34106051D-12	2	73	0	0.10000000D+01
2	49	0	0.10000000D+01	2	73	1	0.15458351D-19
2	49	1	0.17408297D-12	2	74	0	0.10000000D+01
2	50	0	0.10000000D+01	2	74	1	0.78350548D-20
2	50	1	0.88817842D-13	2	75	0	0.10000000D+01
2	51	0	0.10000000D+01	2	75	1	0.39704669D-20
2	51	1	0.45297099D-13	2	76	0	0.10000000D+01
2	52	0	0.10000000D+01	2	76	1	0.20117032D-20
2	52	1	0.23092639D-13	2	77	0	0.10000000D+01
2	53	0	0.10000000D+01	2	77	1	0.10190865D-20
2	53	1	0.11768364D-13	2	78	0	0.10000000D+01
2	54	0	0.10000000D+01	2	78	1	0.51616070D-21
2	54	1	0.59952043D-14	2	79	0	0.10000000D+01
2	55	0	0.10000000D+01	2	79	1	0.26138907D-21
2	55	1	0.30531133D-14	2	80	0	0.10000000D+01
2	56	0	0.10000000D+01	2	80	1	0.13234890D-21
2	56	1	0.15543122D-14	2	81	0	0.10000000D+01
2	57	0	0.10000000D+01	2	81	1	0.67001630D-22
2	57	1	0.79103391D-15	2	82	0	0.10000000D+01
2	58	0	0.10000000D+01	2	82	1	0.33914405D-22
2	58	1	0.40245585D-15	2	83	0	0.10000000D+01
2	59	0	0.10000000D+01	2	83	1	0.17163998D-22
2	59	1	0.20469737D-15	2	84	0	0.10000000D+01
2	60	0	0.10000000D+01	2	84	1	0.86853964D-23
2	60	1	0.10408341D-15	2	85	0	0.10000000D+01
2	61	0	0.10000000D+01	2	85	1	0.43943970D-23
2	61	1	0.52909066D-16	2	86	0	0.10000000D+01
2	62	0	0.10000000D+01	2	86	1	0.22230479D-23
2	62	1	0.26888214D-16	2	87	0	0.10000000D+01
2	63	0	0.10000000D+01	2	87	1	0.11244486D-23
2	63	1	0.13660947D-16	2	88	0	0.10000000D+01
2	64	0	0.10000000D+01	2	88	1	0.56868667D-24
2	64	1	0.69388939D-17	2	89	0	0.10000000D+01
2	65	0	0.10000000D+01	2	89	1	0.28757451D-24
2	65	1	0.35236571D-17	2	90	0	0.10000000D+01
2	66	0	0.10000000D+01	2	90	1	0.14540284D-24
2	66	1	0.17889336D-17	2	91	0	0.10000000D+01
2	67	0	0.10000000D+01	2	91	1	0.73509215D-25
2	67	1	0.90801932D-18	2	92	0	0.10000000D+01
2	68	0	0.10000000D+01	2	92	1	0.37158504D-25
2	68	1	0.46078592D-18	2	93	0	0.10000000D+01
2	69	0	0.10000000D+01	2	93	1	0.18781200D-25
2	69	1	0.23378109D-18	2	94	0	0.10000000D+01
2	70	0	0.10000000D+01	2	94	1	0.94915744D-26
2	70	1	0.11858461D-18	2	95	0	0.10000000D+01

TABLE V

Y_{ij} (α = 2) (CONT'D.)

2	95	1	0.47962743D-26	2	120	0	0.10000000D+01
2	96	0	0.10000000D+01	2	120	1	0.18055593D-33
2	96	1	0.24233807D-26	2	121	0	0.10000000D+01
2	97	0	0.10000000D+01	2	121	1	0.91030283D-34
2	97	1	0.12243121D-26	2	122	0	0.10000000D+01
2	98	0	0.10000000D+01	2	122	1	0.45891299D-34
2	98	1	0.61846695D-27	2	123	0	0.10000000D+01
2	99	0	0.10000000D+01	2	123	1	0.23133729D-34
2	99	1	0.31238892D-27	2	124	0	0.10000000D+01
2	100	0	0.10000000D+01	2	124	1	0.11660904D-34
2	100	1	0.15777218D-27	2	125	0	0.10000000D+01
2	101	0	0.10000000D+01	2	125	1	0.58774718D-35
2	101	1	0.79674951D-28	2	126	0	0.10000000D+01
2	102	0	0.10000000D+01	2	126	1	0.29622458D-35
2	102	1	0.40231906D-28	2	127	0	0.10000000D+01
2	103	0	0.10000000D+01	2	127	1	0.14928778D-35
2	103	1	0.20313168D-28	2	128	0	0.10000000D+01
2	104	0	0.10000000D+01	2	128	1	0.75231638D-36
2	104	1	0.10255192D-28	2	129	0	0.10000000D+01
2	105	0	0.10000000D+01	2	129	1	0.37909693D-36
2	105	1	0.51768997D-29	2	130	0	0.10000000D+01
2	106	0	0.10000000D+01	2	130	1	0.19101783D-36
2	106	1	0.26131017D-29	2	131	0	0.10000000D+01
2	107	0	0.10000000D+01	2	131	1	0.96243600D-37
2	107	1	0.13188768D-29	2	132	0	0.10000000D+01
2	108	0	0.10000000D+01	2	132	1	0.48489142D-37
2	108	1	0.66560139D-30	2	133	0	0.10000000D+01
2	109	0	0.10000000D+01	2	133	1	0.24428242D-37
2	109	1	0.33588218D-30	2	134	0	0.10000000D+01
2	110	0	0.10000000D+01	2	134	1	0.12305956D-37
2	110	1	0.16948184D-30	2	135	0	0.10000000D+01
2	111	0	0.10000000D+01	2	135	1	0.61988960D-38
2	111	1	0.85511290D-31	2	136	0	0.10000000D+01
2	112	0	0.10000000D+01	2	136	1	0.31224069D-38
2	112	1	0.43140831D-31	2	137	0	0.10000000D+01
2	113	0	0.10000000D+01	2	137	1	0.15726829D-38
2	113	1	0.21763008D-31	2	138	0	0.10000000D+01
2	114	0	0.10000000D+01	2	138	1	0.79208115D-39
2	114	1	0.10977801D-31	2	139	0	0.10000000D+01
2	115	0	0.10000000D+01	2	139	1	0.39891044D-39
2	115	1	0.55370486D-32	2	140	0	0.10000000D+01
2	116	0	0.10000000D+01	2	140	1	0.20089015D-39
2	116	1	0.27925984D-32	2	141	0	0.10000000D+01
2	117	0	0.10000000D+01	2	141	1	0.10116254D-39
2	117	1	0.14083363D-32	2	142	0	0.10000000D+01
2	118	0	0.10000000D+01	2	142	1	0.50940002D-40
2	118	1	0.71018667D-33	2	143	0	0.10000000D+01
2	119	0	0.10000000D+01	2	143	1	0.25649367D-40
2	119	1	0.35810260D-33	2	144	0	0.10000000D+01

TABLE V

Y_{ij} (α = 2) (CONT'D.)

2	144	1	0.12914367D-40
2	145	0	0.10000000D+01
2	145	1	0.65020249D-41
2	146	0	0.10000000D+01
2	146	1	0.32734332D-41
2	147	0	0.10000000D+01
2	147	1	0.16479270D-41
2	148	0	0.10000000D+01
2	148	1	0.82956869D-42
2	149	0	0.10000000D+01
2	149	1	0.41758694D-42
2	150	0	0.10000000D+01
2	150	1	0.21019477D-42

TABLE VI

Y_{IJ} (a = 3)

a	i	j	Y_{ij}	a	i	j	Y_{ij}
3	1	1	0.10000000D+01	3	17	0	0.97412486D+00
3	2	0	0.33333329D+00	3	17	1	0.25868823D-01
3	2	2	0.66666671D+00	3	17	2	-0.63187159D-05
3	3	0	0.11111124D+00	3	18	0	0.98173310D+00
3	3	1	0.66666663D+00	3	18	1	0.18264527D-01
3	3	3	0.22222213D+00	3	18	2	0.23695185D-05
3	4	0	0.25925926D+00	3	19	0	0.98714473D+00
3	4	1	0.29629630D+00	3	19	1	0.12854385D-01
3	4	2	0.44444444D+00	3	19	2	0.88276178D-06
3	5	0	0.25925926D+00	3	20	0	0.99097847D+00
3	5	1	0.49382716D+00	3	20	1	0.90212059D-02
3	5	2	0.24691358D+00	3	20	2	0.32694881D-06
3	6	0	0.33333333D+00	3	21	0	0.99368482D+00
3	6	1	0.54320988D+00	3	21	1	0.63150609D-02
3	6	2	0.12345679D+00	3	21	2	0.12045482D-06
3	7	0	0.44307270D+00	3	22	0	0.99558936D+00
3	7	1	0.49931413D+00	3	22	1	0.44105987D-02
3	7	2	0.57613169D-01	3	22	2	0.44166769D-07
3	8	0	0.55738455D+00	3	23	0	0.99692590D+00
3	8	1	0.41700960D+00	3	23	1	0.30740829D-02
3	8	2	0.25605853D-01	3	23	2	0.16124376D-07
3	9	0	0.65980796D+00	3	24	0	0.99786149D+00
3	9	1	0.32921811D+00	3	24	1	0.21385032D-02
3	9	2	0.10973937D-01	3	24	2	0.58634094D-08
3	10	0	0.74444952D+00	3	25	0	0.99851492D+00
3	10	1	0.25097800D+00	3	25	1	0.14850756D-02
3	10	2	0.45724737D-02	3	25	2	0.21244237D-08
3	11	0	0.81110603D+00	3	26	0	0.99897035D+00
3	11	1	0.18703111D+00	3	26	1	0.10296538D-02
3	11	2	0.18628597D-02	3	26	2	0.76715300D-09
3	12	0	0.86201290D+00	3	27	0	0.99928716D+00
3	12	1	0.13724195D+00	3	27	1	0.71283775D-03
3	12	2	0.74514386D-03	3	27	2	0.27617508D-09
3	13	0	0.90009804D+00	3	28	0	0.99950717D+00
3	13	1	0.99608423D-01	3	28	1	0.49282628D-03
3	13	2	0.29354152D-03	3	28	2	0.99139773D-10
3	14	0	0.92817892D+00	3	29	0	0.99965972D+00
3	14	1	0.71706925D-01	3	29	1	0.34028488D-03
3	14	2	0.11415504D-03	3	29	2	0.35494486D-10
3	15	0	0.94866159D+00	3	30	0	0.99976532D+00
3	15	1	0.51294499D-01	3	30	1	0.23467925D-03
3	15	2	0.43905783D-04	3	30	2	0.12676602D-10
3	16	0	0.96347819D+00	3	31	0	0.99983833D+00
3	16	1	0.36505080D-01	3	31	1	0.16166794D-03
3	16	2	0.16726013D-04	3	31	2	0.45169502D-11

TABLE VI

Y_{ij} (α = 3) (CONT'D.)

3	32	0	0.99988874D+00	3	48	1	0.25406922D-06
3	32	1	0.11125536D-03	3	48	2	0.84847710D-19
3	32	2	0.16060268D-11	3	49	0	0.99999983D+00
3	33	0	0.99992351D+00	3	49	1	0.17290822D-06
3	33	1	0.76488060D-04	3	49	2	0.29486083D-19
3	33	2	0.56988046D-12	3	50	0	0.99999988D+00
3	34	0	0.99994746D+00	3	50	1	0.11762464D-06
3	34	1	0.52537253D-04	3	50	2	0.10238223D-19
3	34	2	0.20183266D-12	3	51	0	0.99999992D+00
3	35	0	0.99996395D+00	3	51	1	0.79984756D-07
3	35	1	0.36054978D-04	3	51	2	0.35520367D-20
3	35	2	0.71354982D-13	3	52	0	0.99999995D+00
3	36	0	0.99997528D+00	3	52	1	0.54368723D-07
3	36	1	0.24723414D-04	3	52	2	0.12313727D-20
3	36	2	0.25184111D-13	3	53	0	0.99999996D+00
3	37	0	0.99998306D+00	3	53	1	0.36942850D-07
3	37	1	0.16940117D-04	3	53	2	0.42655395D-21
3	37	2	0.88744011D-14	3	54	0	0.99999997D+00
3	38	0	0.99998840D+00	3	54	1	0.25093257D-07
3	38	1	0.11598638D-04	3	54	2	0.14765329D-21
3	38	2	0.31224745D-14	3	55	0	0.99999998D+00
3	39	0	0.99999206D+00	3	55	1	0.17038631D-07
3	39	1	0.79359105D-05	3	55	2	0.51075038D-22
3	39	2	0.10970856D-14	3	56	0	0.99999999D+00
3	40	0	0.99999457D+00	3	56	1	0.11565616D-07
3	40	1	0.54262636D-05	3	56	2	0.17655569D-22
3	40	2	0.38494232D-15	3	57	0	0.99999999D+00
3	41	0	0.99999629D+00	3	57	1	0.78480968D-08
3	41	1	0.37079468D-05	3	57	2	0.60991964D-23
3	41	2	0.13489432D-15	3	58	0	0.99999999D+00
3	42	0	0.99999747D+00	3	58	1	0.53238551D-08
3	42	1	0.25322564D-05	3	58	2	0.21056749D-23
3	42	2	0.47213012D-16	3	59	0	0.10000000D+01
3	43	0	0.99999827D+00	3	59	1	0.36104305D-08
3	43	1	0.17283654D-05	3	59	2	0.72651943D-24
3	43	2	0.16505362D-16	3	60	0	0.10000000D+01
3	44	0	0.99999882D+00	3	60	1	0.24477495D-08
3	44	1	0.11790400D-05	3	60	2	0.25052394D-24
3	44	2	0.57637771D-17	3	61	0	0.10000000D+01
3	45	0	0.99999920D+00	3	61	1	0.16590302D-08
3	45	1	0.80389091D-06	3	61	2	0.86338759D-25
3	45	2	0.20106199D-17	3	62	0	0.10000000D+01
3	46	0	0.99999945D+00	3	62	1	0.11241516D-08
3	46	1	0.54783677D-06	3	62	2	0.29738906D-25
3	46	2	0.70067058D-18	3	63	0	0.10000000D+01
3	47	0	0.99999963D+00	3	63	1	0.76152206D-09
3	47	1	0.37316417D-06	3	63	2	0.10237984D-25
3	47	2	0.24393717D-18	3	64	0	0.10000000D+01
3	48	0	0.99999975D+00	3	64	1	0.51573981D-09

TABLE VI

Y_{ij} (α = 3) (CONT'D.)

3	64	2	0.35227472D-26	3	81	0	0.10000000D+01
3	65	0	0.10000000D+01	3	81	1	0.66249758D-12
3	65	1	0.34919883D-09	3	81	2	0.43840412D-34
3	65	2	0.12115268D-26	3	82	0	0.10000000D+01
3	66	0	0.10000000D+01	3	82	1	0.44711771D-12
3	66	1	0.23638074D-09	3	82	2	0.14978807D-34
3	66	2	0.41646234D-27	3	83	0	0.10000000D+01
3	67	0	0.10000000D+01	3	83	1	0.30171357D-12
3	67	1	0.15997485D-09	3	83	2	0.51162182D-35
3	67	2	0.14309219D-27	3	84	0	0.10000000D+01
3	68	0	0.10000000D+01	3	84	1	0.20356579D-12
3	68	1	0.10824169D-09	3	84	2	0.17470013D-35
3	68	2	0.49142772D-28	3	85	0	0.10000000D+01
3	69	0	0.10000000D+01	3	85	1	0.13732612D-12
3	69	1	0.73222318D-10	3	85	2	0.59636592D-36
3	69	2	0.16869907D-28	3	86	0	0.10000000D+01
3	70	0	0.10000000D+01	3	86	1	0.92627818D-13
3	70	1	0.49522341D-10	3	86	2	0.20352170D-36
3	70	2	0.57886935D-29	3	87	0	0.10000000D+01
3	71	0	0.10000000D+01	3	87	1	0.62469923D-13
3	71	1	0.33486535D-10	3	87	2	0.69436816D-37
3	71	2	0.19854939D-29	3	88	0	0.10000000D+01
3	72	0	0.10000000D+01	3	88	1	0.42125312D-13
3	72	1	0.22638784D-10	3	88	2	0.23683875D-37
3	72	2	0.68074077D-30	3	89	0	0.10000000D+01
3	73	0	0.10000000D+01	3	89	1	0.28402673D-13
3	73	1	0.15302141D-10	3	89	2	0.80761107D-38
3	73	2	0.23330552D-30	3	90	0	0.10000000D+01
3	74	0	0.10000000D+01	3	90	1	0.19147869D-13
3	74	1	0.10341173D-10	3	90	2	0.27532196D-38
3	74	2	0.79928743D-31	3	91	0	0.10000000D+01
3	75	0	0.10000000D+01	3	91	1	0.12907082D-13
3	75	1	0.69872791D-11	3	91	2	0.93836322D-39
3	75	2	0.27372857D-31	3	92	0	0.10000000D+01
3	76	0	0.10000000D+01	3	92	1	0.86992789D-14
3	76	1	0.47202952D-11	3	92	2	0.31973858D-39
3	76	2	0.93708881D-32	3	93	0	0.10000000D+01
3	77	0	0.10000000D+01	3	93	1	0.58625575D-14
3	77	1	0.31882696D-11	3	93	2	0.10892193D-39
3	77	2	0.32069261D-32	3	94	0	0.10000000D+01
3	78	0	0.10000000D+01	3	94	1	0.39503972D-14
3	78	1	0.21531171D-11	3	94	2	0.37096601D-40
3	78	2	0.10971063D-32	3	95	0	0.10000000D+01
3	79	0	0.10000000D+01	3	95	1	0.26616151D-14
3	79	1	0.14538141D-11	3	95	2	0.12631459D-40
3	79	2	0.37520086D-33	3	96	0	0.10000000D+01
3	80	0	0.10000000D+01	3	96	1	0.17930881D-14
3	80	1	0.98147789D-12	3	96	2	0.43000711D-41
3	80	2	0.12827380D-33	3	97	0	0.10000000D+01

TABLE VI

Y_{ij} (α = 3) (CONT'D.)

3	97	1	0.12078441D-14	3	113	2	0.46207869D-49
3	97	2	0.14635330D-41	3	114	0	0.10000000D+01
3	98	0	0.10000000D+01	3	114	1	0.14407631D-17
3	98	1	0.81353070D-15	3	114	2	0.15677670D-49
3	98	2	0.49800775D-42	3	115	0	0.10000000D+01
3	99	0	0.10000000D+01	3	115	1	0.96893425D-18
3	99	1	0.54788802D-15	3	115	2	0.53183835D-50
3	99	2	0.16942532D-42	3	116	0	0.10000000D+01
3	100	0	0.10000000D+01	3	116	1	0.65157318D-18
3	100	1	0.36894816D-15	3	116	2	0.18038962D-50
3	100	2	0.57627659D-43	3	117	0	0.10000000D+01
3	101	0	0.10000000D+01	3	117	1	0.43812679D-18
3	101	1	0.24842510D-15	3	117	2	0.61175609D-51
3	101	2	0.19597285D-43	3	118	0	0.10000000D+01
3	102	0	0.10000000D+01	3	118	1	0.29458098D-18
3	102	1	0.16725650D-15	3	118	2	0.20743454D-51
3	102	2	0.66630768D-44	3	119	0	0.10000000D+01
3	103	0	0.10000000D+01	3	119	1	0.19805162D-18
3	103	1	0.11259751D-15	3	119	2	0.70326809D-52
3	103	2	0.22650063D-44	3	120	0	0.10000000D+01
3	104	0	0.10000000D+01	3	120	1	0.13314394D-18
3	104	1	0.75793796D-16	3	120	2	0.23839596D-52
3	104	2	0.76980606D-45	3	121	0	0.10000000D+01
3	105	0	0.10000000D+01	3	121	1	0.89502318D-19
3	105	1	0.51015055D-16	3	121	2	0.80800872D-53
3	105	2	0.26158458D-45	3	122	0	0.10000000D+01
3	106	0	0.10000000D+01	3	122	1	0.60161338D-19
3	106	1	0.34333942D-16	3	122	2	0.27382518D-53
3	106	2	0.88871686D-46	3	123	0	0.10000000D+01
3	107	0	0.10000000D+01	3	123	1	0.40436309D-19
3	107	1	0.23105231D-16	3	123	2	0.92783738D-54
3	107	2	0.30188160D-46	3	124	0	0.10000000D+01
3	108	0	0.10000000D+01	3	124	1	0.27176706D-19
3	108	1	0.15547445D-16	3	124	2	0.31434928D-54
3	108	2	0.10252583D-46	3	125	0	0.10000000D+01
3	109	0	0.10000000D+01	3	125	1	0.18263916D-19
3	109	1	0.10460935D-16	3	125	2	0.10648688D-54
3	109	2	0.34814066D-47	3	126	0	0.10000000D+01
3	110	0	0.10000000D+01	3	126	1	0.12273351D-19
3	110	1	0.70379382D-17	3	126	2	0.36068138D-55
3	110	2	0.11819590D-47	3	127	0	0.10000000D+01
3	111	0	0.10000000D+01	3	127	1	0.82471725D-20
3	111	1	0.47346130D-17	3	127	2	0.12215076D-55
3	111	2	0.40121545D-48	3	128	0	0.10000000D+01
3	112	0	0.10000000D+01	3	128	1	0.55414073D-20
3	112	1	0.31848448D-17	3	128	2	0.41363220D-56
3	112	2	0.13617009D-48	3	129	0	0.10000000D+01
3	113	0	0.10000000D+01	3	129	1	0.37231330D-20
3	113	1	0.21421872D-17	3	129	2	0.14004870D-56

TABLE VI

Y_{ij} (α = 3) (CONT'D.)

3	130	0	0.10000000D+01
3	130	1	0.25013297D-20
3	130	2	0.47412320D-57
3	131	0	0.10000000D+01
3	131	1	0.16803804D-20
3	131	2	0.16049131D-57
3	132	0	0.10000000D+01
3	132	1	0.11288052D-20
3	132	2	0.54320137D-58
3	133	0	0.10000000D+01
3	133	1	0.75823783D-21
3	133	2	0.18383151D-58
3	134	0	0.10000000D+01
3	134	1	0.50929257D-21
3	134	2	0.62205611D-59
3	135	0	0.10000000D+01
3	135	1	0.34206218D-21
3	135	2	0.21047011D-59
3	136	0	0.10000000D+01
3	136	1	0.22973065D-21
3	136	2	0.71203819D-60
3	137	0	0.10000000D+01
3	137	1	0.15427990D-21
3	137	2	0.24086230D-60
3	138	0	0.10000000D+01
3	138	1	0.10360402D-21
3	138	2	0.81468132D-61
3	139	0	0.10000000D+01
3	139	1	0.69569847D-22
3	139	2	0.27552483D-61
3	140	0	0.10000000D+01
3	140	1	0.46713567D-22
3	140	2	0.93172646D-62
3	141	0	0.10000000D+01
3	141	1	0.31364823D-22
3	141	2	0.31504420D-62
3	142	0	0.10000000D+01
3	142	1	0.21058179D-22
3	142	2	0.10651494D-62
3	143	0	0.10000000D+01
3	143	1	0.14137651D-22
3	143	2	0.36008598D-63
3	144	0	0.10000000D+01
3	144	1	0.94910104D-23
3	144	2	0.12171920D-63
3	145	0	0.10000000D+01
3	145	1	0.63712801D-23
3	145	2	0.41140524D-64
3	146	0	0.10000000D+01
3	146	1	0.42768133D-23
3	146	2	0.13903973D-64
3	147	0	0.10000000D+01
3	147	1	0.28707377D-23
3	147	2	0.46985841D-65
3	148	0	0.10000000D+01
3	148	1	0.19268444D-23
3	148	2	0.15876494D-65
3	149	0	0.10000000D+01
3	149	1	0.12932424D-23
3	149	2	0.53641669D-66
3	150	0	0.10000000D+01
3	150	1	0.86794791D-24
3	150	2	0.18122186D-66

TABLE VII

Y_{IJ} (α = 5)

α	i	j	Y_{ij}	α	i	j	Y_{ij}
5	1	0	0.00000000D+00	5	11	0	0.21120913D+00
5	1	1	0.10000000D+01	5	11	1	0.44674827D+00
5	2	0	0.20000000D+00	5	11	2	0.29257112D+00
5	2	1	0.00000000D+00	5	11	3	0.48660477D-01
5	2	2	0.80000000D+00	5	11	4	0.81100800D-03
5	3	0	0.40000219D-01	5	12	0	0.26103017D+00
5	3	1	0.48000000D+00	5	12	1	0.47434335D+00
5	3	2	0.00000000D+00	5	12	2	0.23767399D+00
5	3	3	0.47999978D+00	5	12	3	0.26709195D-01
5	4	0	0.10399974D+00	5	12	4	0.24330240D-03
5	4	1	0.12800045D+00	5	13	0	0.31870761D+00
5	4	2	0.57599989D+00	5	13	1	0.48348588D+00
5	4	3	0.00000000D+00	5	13	2	0.18362252D+00
5	4	4	0.19199992D+00	5	13	3	0.14113701D-01
5	5	0	0.65600254D-01	5	13	4	0.70287360D-04
5	5	1	0.31999977D+00	5	14	0	0.38150301D+00
5	5	2	0.19200021D+00	5	14	1	0.47462477D+00
5	5	3	0.38399978D+00	5	14	2	0.13660296D+00
5	5	4	0.00000000D+00	5	14	3	0.72495657D-02
5	5	5	0.38399986D-01	5	14	4	0.19680461D-04
5	6	0	0.89919848D-01	5	15	0	0.44634408D+00
5	6	1	0.23808024D+00	5	15	1	0.45125828D+00
5	6	2	0.40319998D+00	5	15	2	0.98749596D-01
5	6	3	0.15359999D+00	5	15	3	0.36426742D-02
5	6	4	0.11519995D+00	5	15	4	0.53673984D-05
5	7	0	0.95039994D-01	5	16	0	0.51047719D+00
5	7	1	0.29747204D+00	5	16	1	0.41789669D+00
5	7	2	0.33868798D+00	5	16	2	0.69826522D-01
5	7	3	0.21503999D+00	5	16	3	0.17981609D-02
5	7	4	0.53760000D-01	5	16	4	0.14313062D-05
5	8	0	0.11362560D+00	5	17	0	0.57177287D+00
5	8	1	0.32727043D+00	5	17	1	0.37882223D+00
5	8	2	0.34836478D+00	5	17	2	0.48529849D-01
5	8	3	0.18923519D+00	5	17	3	0.87467591D-03
5	8	4	0.21504000D-01	5	17	4	0.37434163D-06
5	9	0	0.13885697D+00	5	18	0	0.62878595D+00
5	9	1	0.36200450D+00	5	18	1	0.33752450D+00
5	9	2	0.35721214D+00	5	18	2	0.33269271D-01
5	9	3	0.13418495D+00	5	18	3	0.42017813D-03
5	9	4	0.77414400D-02	5	18	4	0.96259277D-07
5	10	0	0.17069313D+00	5	19	0	0.68068236D+00
5	10	1	0.40564738D+00	5	19	1	0.29656144D+00
5	10	2	0.33702910D+00	5	19	2	0.22556512D-01
5	10	3	0.84049915D-01	5	19	3	0.19966996D-03
5	10	4	0.25804800D-02	5	19	4	0.24385683D-07

TABLE VII

Y_{ij} (a = 5) (CONT'D.)

5	20	0	0.72711274D+00	5	29	4	0.15301465D-13
5	20	1	0.25763815D+00	5	30	0	0.95379092D+00
5	20	2	0.15155124D-01	5	30	1	0.45995447D-01
5	20	3	0.93983853D-04	5	30	2	0.21359946D-03
5	20	4	0.60964209D-08	5	30	3	0.35106444D-07
5	21	0	0.76807986D+00	5	30	4	0.35311074D-14
5	21	1	0.22177063D+00	5	31	0	0.96176091D+00
5	21	2	0.10105641D-01	5	31	1	0.38102056D-01
5	21	3	0.43864487D-04	5	31	2	0.13701918D-03
5	21	4	0.15061746D-08	5	31	3	0.15547142D-07
5	22	0	0.80382208D+00	5	31	4	0.81084688D-15
5	22	1	0.18946198D+00	5	32	0	0.96839645D+00
5	22	2	0.66956257D-02	5	32	1	0.31515841D-01
5	22	3	0.20317501D-04	5	32	2	0.87701539D-04
5	22	4	0.36817601D-09	5	32	3	0.68621876D-08
5	23	0	0.83472065D+00	5	32	4	0.18533643D-15
5	23	1	0.16085801D+00	5	33	0	0.97391073D+00
5	23	2	0.44119989D-02	5	33	1	0.26033245D-01
5	23	3	0.93463713D-05	5	33	2	0.56019913D-04
5	23	4	0.89137349D-10	5	33	3	0.30193627D-08
5	24	0	0.86123094D+00	5	33	4	0.42180015D-16
5	24	1	0.13587140D+00	5	34	0	0.97848566D+00
5	24	2	0.28933888D-02	5	34	1	0.21478627D-01
5	24	3	0.42727043D-05	5	34	2	0.35714495D-04
5	24	4	0.21392964D-10	5	34	3	0.13246237D-08
5	25	0	0.88383425D+00	5	34	4	0.95608034D-17
5	25	1	0.11427428D+00	5	35	0	0.98227562D+00
5	25	2	0.18895259D-02	5	35	1	0.17701650D-01
5	25	3	0.19421568D-05	5	35	2	0.22728196D-04
5	25	4	0.50935628D-11	5	35	3	0.57952287D-09
5	26	0	0.90300588D+00	5	35	4	0.21588911D-17
5	26	1	0.95763899D-01	5	36	0	0.98541111D+00
5	26	2	0.12293445D-02	5	36	1	0.14574455D-01
5	26	3	0.87819706D-06	5	36	2	0.14439435D-04
5	26	4	0.12039330D-11	5	36	3	0.25288271D-09
5	27	0	0.91919543D+00	5	36	4	0.48575049D-18
5	27	1	0.80007033D-01	5	37	0	0.98800197D+00
5	27	2	0.79713687D-03	5	37	1	0.11988868D-01
5	27	3	0.39518971D-06	5	37	2	0.91588784D-05
5	27	4	0.28266254D-12	5	37	3	0.11007836D-09
5	28	0	0.93281582D+00	5	37	4	0.10892587D-18
5	28	1	0.66668699D-01	5	38	0	0.99014044D+00
5	28	2	0.51530798D-03	5	38	1	0.98537625D-02
5	28	3	0.17704523D-06	5	38	2	0.58006887D-05
5	28	4	0.65954592D-13	5	38	3	0.47805457D-10
5	29	0	0.94423819D+00	5	38	4	0.24348135D-19
5	29	1	0.55429536D-01	5	39	0	0.99190369D+00
5	29	2	0.33219268D-03	5	39	1	0.80926459D-02
5	29	3	0.78989469D-07	5	39	2	0.36685722D-05

TABLE VII

Y_{ij} ($\alpha = 5$) (CONT'D.)

5	39	3	0.20715698D-10	5	49	2	0.35205125D-07
5	39	4	0.54261559D-20	5	49	3	0.43790987D-14
5	40	0	0.99335618D+00	5	49	4	0.14313070D-26
5	40	1	0.66415059D-02	5	50	0	0.99910799D+00
5	40	2	0.23170054D-05	5	50	1	0.89198580D-03
5	40	3	0.89581398D-11	5	50	2	0.22003206D-07
5	40	4	0.12058124D-20	5	50	3	0.18634463D-14
5	41	0	0.99455163D+00	5	50	4	0.31115370D-27
5	41	1	0.54469118D-02	5	51	0	0.99927212D+00
5	41	2	0.14615010D-05	5	51	1	0.72786884D-03
5	41	3	0.38661445D-11	5	51	2	0.13740778D-07
5	41	4	0.26723410D-21	5	51	3	0.79196466D-15
5	42	0	0.99553471D+00	5	51	4	0.67526973D-28
5	42	1	0.44643646D-02	5	52	0	0.99940627D+00
5	42	2	0.92074797D-06	5	52	1	0.59371789D-03
5	42	3	0.16654161D-11	5	52	2	0.85742463D-08
5	42	4	0.59072802D-22	5	52	3	0.33618092D-15
5	43	0	0.99634254D+00	5	52	4	0.14630844D-28
5	43	1	0.36568766D-02	5	53	0	0.99951588D+00
5	43	2	0.57939850D-06	5	53	1	0.48411172D-03
5	43	3	0.71612893D-12	5	53	2	0.53462949D-08
5	43	4	0.13026310D-22	5	53	3	0.14254071D-15
5	44	0	0.99700588D+00	5	53	4	0.31650397D-29
5	44	1	0.29937564D-02	5	54	0	0.99960540D+00
5	44	2	0.36419377D-06	5	54	1	0.39459878D-03
5	44	3	0.30741144D-12	5	54	2	0.33311531D-08
5	44	4	0.28657882D-23	5	54	3	0.60370183D-16
5	45	0	0.99755020D+00	5	54	4	0.68364859D-30
5	45	1	0.24495757D-02	5	55	0	0.99967847D+00
5	45	2	0.22867999D-06	5	55	1	0.32152621D-03
5	45	3	0.13174776D-12	5	55	2	0.20741142D-08
5	45	4	0.62907546D-24	5	55	3	0.25541231D-16
5	46	0	0.99799656D+00	5	55	4	0.14745362D-30
5	46	1	0.20032957D-02	5	56	0	0.99973810D+00
5	46	2	0.14344480D-06	5	56	1	0.26189851D-03
5	46	3	0.56375786D-13	5	56	2	0.12905600D-08
5	46	4	0.13779748D-24	5	56	3	0.10794785D-16
5	47	0	0.99836238D+00	5	56	4	0.31759240D-31
5	47	1	0.16375312D-02	5	57	0	0.99978674D+00
5	47	2	0.89892110D-07	5	57	1	0.21326071D-03
5	47	3	0.24087836D-13	5	57	2	0.80249366D-09
5	47	4	0.30123171D-25	5	57	3	0.45577979D-17
5	48	0	0.99866201D+00	5	57	4	0.68312329D-32
5	48	1	0.13379322D-02	5	58	0	0.99982640D+00
5	48	2	0.56280292D-07	5	58	1	0.17360201D-03
5	48	3	0.10277477D-13	5	58	2	0.49869249D-09
5	48	4	0.65723281D-26	5	58	3	0.19225620D-17
5	49	0	0.99890730D+00	5	58	4	0.14674500D-32
5	49	1	0.10926661D-02	5	59	0	0.99985872D+00

TABLE VII

Y_{ij} (α = 5) (CONT'D.)

5	59	1	0.14127631D-03	5	69	0	0.99998226D+00
5	59	2	0.30971429D-09	5	69	1	0.17740591D-04
5	59	3	0.81022257D-18	5	69	2	0.25677441D-11
5	59	4	0.31483473D-33	5	69	3	0.13692482D-21
5	60	0	0.99988506D+00	5	69	4	0.61237358D-40
5	60	1	0.11493678D-03	5	70	0	0.99998560D+00
5	60	2	0.19223645D-09	5	70	1	0.14398162D-04
5	60	3	0.34114634D-18	5	70	2	0.15859596D-11
5	60	4	0.67464585D-34	5	70	3	0.57222315D-22
5	61	0	0.99990652D+00	5	70	4	0.12989743D-40
5	61	1	0.93481989D-04	5	71	0	0.99998832D+00
5	61	2	0.11925177D-09	5	71	1	0.11683080D-04
5	61	3	0.14351674D-18	5	71	2	0.97915767D-12
5	61	4	0.14439788D-34	5	71	3	0.23898732D-22
5	62	0	0.99992399D+00	5	71	4	0.27530499D-41
5	62	1	0.76011631D-04	5	72	0	0.99999052D+00
5	62	2	0.73936095D-10	5	72	1	0.94781050D-05
5	62	3	0.60325680D-19	5	72	2	0.60428016D-12
5	62	4	0.30871272D-35	5	72	3	0.99751227D-23
5	63	0	0.99993821D+00	5	72	4	0.58299881D-42
5	63	1	0.61790129D-04	5	73	0	0.99999231D+00
5	63	2	0.45816138D-10	5	73	1	0.76877965D-05
5	63	3	0.25336785D-19	5	73	2	0.37278128D-12
5	63	4	0.65928479D-36	5	73	3	0.41610512D-23
5	64	0	0.99994978D+00	5	73	4	0.12335917D-42
5	64	1	0.50216757D-04	5	74	0	0.99999377D+00
5	64	2	0.28376447D-10	5	74	1	0.62344872D-05
5	64	3	0.10633143D-19	5	74	2	0.22988179D-12
5	64	4	0.14064742D-36	5	74	3	0.17347481D-23
5	65	0	0.99995920D+00	5	74	4	0.26081653D-43
5	65	1	0.40801126D-04	5	75	0	0.99999495D+00
5	65	2	0.17566372D-10	5	75	1	0.50549897D-05
5	65	3	0.44590599D-20	5	75	2	0.14170795D-12
5	65	4	0.29974041D-37	5	75	3	0.72281171D-24
5	66	0	0.99996686D+00	5	75	4	0.55102083D-44
5	66	1	0.33143075D-04	5	76	0	0.99999590D+00
5	66	2	0.10869193D-10	5	76	1	0.40979117D-05
5	66	3	0.18685584D-20	5	76	2	0.87322739D-13
5	66	4	0.63815699D-38	5	76	3	0.30100652D-24
5	67	0	0.99997308D+00	5	76	4	0.11632662D-44
5	67	1	0.26916199D-04	5	77	0	0.99999668D+00
5	67	2	0.67221775D-11	5	77	1	0.33214653D-05
5	67	3	0.78245884D-21	5	77	2	0.53790807D-13
5	67	4	0.13573498D-38	5	77	3	0.12528379D-24
5	68	0	0.99997815D+00	5	77	4	0.24540136D-45
5	68	1	0.21854349D-04	5	78	0	0.99999731D+00
5	68	2	0.41555279D-11	5	78	1	0.26916810D-05
5	68	3	0.32742893D-21	5	78	2	0.33123813D-13
5	68	4	0.28843683D-39	5	78	3	0.52118059D-25

TABLE VII

Y_{ij} (a = 5) (CONT'D.)

5	78	4	0.51733260D-46	5	88	3	0.78829650D-29
5	79	0	0.99999782D+00	5	88	4	0.86602200D-53
5	79	1	0.21809518D-05	5	89	0	0.99999974D+00
5	79	2	0.20390503D-13	5	89	1	0.26382070D-06
5	79	3	0.21670140D-25	5	89	2	0.15670809D-15
5	79	4	0.10898474D-46	5	89	3	0.32631809D-29
5	80	0	0.99999823D+00	5	89	4	0.18135520D-53
5	80	1	0.17668470D-05	5	90	0	0.99999979D+00
5	80	2	0.12548002D-13	5	90	1	0.21342798D-06
5	80	3	0.90057725D-26	5	90	2	0.96161785D-16
5	80	4	0.22944155D-47	5	90	3	0.13502817D-29
5	81	0	0.99999857D+00	5	90	4	0.37958064D-54
5	81	1	0.14311461D-05	5	91	0	0.99999983D+00
5	81	2	0.77194036D-14	5	91	1	0.17263952D-06
5	81	3	0.37408594D-26	5	91	2	0.58993634D-16
5	81	4	0.48272118D-48	5	91	3	0.55852563D-30
5	82	0	0.99999884D+00	5	91	4	0.79406525D-55
5	82	1	0.11590517D-05	5	92	0	0.99999986D+00
5	82	2	0.47474332D-14	5	92	1	0.13962933D-06
5	82	3	0.15531669D-26	5	92	2	0.36182762D-16
5	82	4	0.10149522D-48	5	92	3	0.23094093D-30
5	83	0	0.99999906D+00	5	92	4	0.16603182D-55
5	83	1	0.93854915D-06	5	93	0	0.99999989D+00
5	83	2	0.29187923D-14	5	93	1	0.11291763D-06
5	83	3	0.64456427D-27	5	93	2	0.22186793D-16
5	83	4	0.21326844D-49	5	93	3	0.95455586D-31
5	84	0	0.99999924D+00	5	93	4	0.34698786D-56
5	84	1	0.75988558D-06	5	94	0	0.99999991D+00
5	84	2	0.17939894D-14	5	94	1	0.91305438D-07
5	84	3	0.26737481D-27	5	94	2	0.13601469D-16
5	84	4	0.44786373D-50	5	94	3	0.39440989D-31
5	85	0	0.99999938D+00	5	94	4	0.72481908D-57
5	85	1	0.61514547D-06	5	95	0	0.99999993D+00
5	85	2	0.11023308D-14	5	95	1	0.73821418D-07
5	85	3	0.11086273D-27	5	95	2	0.83363840D-17
5	85	4	0.93996091D-51	5	95	3	0.16290843D-31
5	86	0	0.99999950D+00	5	95	4	0.15133585D-57
5	86	1	0.49790598D-06	5	96	0	0.99999994D+00
5	86	2	0.67714609D-15	5	96	1	0.59678788D-07
5	86	3	0.45947925D-28	5	96	2	0.51082523D-17
5	86	4	0.19716253D-51	5	96	3	0.67265418D-32
5	87	0	0.99999960D+00	5	96	4	0.31583134D-58
5	87	1	0.40295647D-06	5	97	0	0.99999995D+00
5	87	2	0.41584736D-15	5	97	1	0.48240354D-07
5	87	3	0.19035569D-28	5	97	2	0.31294767D-17
5	87	4	0.41332868D-52	5	97	3	0.27764875D-32
5	88	0	0.99999967D+00	5	97	4	0.65883098D-59
5	88	1	0.32607052D-06	5	98	0	0.99999996D+00
5	88	2	0.25531094D-15	5	98	1	0.38990142D-07

TABLE VII

Y_{ij} (α = 5) (CONT'D.)

5	98	2	0.19168045D-17	5	108	1	0.46137320D-08
5	98	3	0.11456664D-32	5	108	2	0.14089645D-19
5	98	4	0.13737327D-59	5	108	3	0.16125101D-36
5	99	0	0.99999997D+00	5	108	4	0.20869509D-66
5	99	1	0.31510400D-07	5	109	0	0.10000000D+01
5	99	2	0.11737957D-17	5	109	1	0.37251614D-08
5	99	3	0.47258739D-33	5	109	2	0.86118015D-20
5	99	4	0.28631481D-60	5	109	3	0.66325888D-37
5	100	0	0.99999997D+00	5	109	4	0.43329076D-67
5	100	1	0.25462950D-07	5	110	0	0.10000000D+01
5	100	2	0.71865044D-18	5	110	1	0.30074697D-08
5	100	3	0.19488140D-33	5	110	2	0.52627676D-20
5	100	4	0.59648919D-61	5	110	3	0.27274197D-37
5	101	0	0.99999998D+00	5	110	4	0.89928271D-68
5	101	1	0.20574063D-07	5	111	0	0.10000000D+01
5	101	2	0.43990118D-18	5	111	1	0.24278483D-08
5	101	3	0.80338863D-34	5	111	2	0.32155993D-20
5	101	4	0.12421734D-61	5	111	3	0.11212725D-37
5	102	0	0.99999998D+00	5	111	4	0.18658015D-68
5	102	1	0.16622214D-07	5	112	0	0.10000000D+01
5	102	2	0.26921952D-18	5	112	1	0.19597766D-08
5	102	3	0.33109349D-34	5	112	2	0.19644388D-20
5	102	4	0.25857486D-62	5	112	3	0.46085330D-38
5	103	0	0.99999999D+00	5	112	4	0.38698105D-69
5	103	1	0.13428141D-07	5	113	0	0.10000000D+01
5	103	2	0.16473036D-18	5	113	1	0.15818197D-08
5	103	3	0.13641052D-34	5	113	2	0.11999005D-20
5	103	4	0.53804466D-63	5	113	3	0.18936881D-38
5	104	0	0.99999999D+00	5	113	4	0.80236439D-70
5	104	1	0.10846809D-07	5	114	0	0.10000000D+01
5	104	2	0.10077622D-18	5	114	1	0.12766545D-08
5	104	3	0.56184927D-35	5	114	2	0.73279636D-21
5	104	4	0.11191329D-63	5	114	3	0.77794755D-39
5	105	0	0.99999999D+00	5	114	4	0.16630825D-70
5	105	1	0.87608843D-08	5	115	0	0.10000000D+01
5	105	2	0.61639824D-19	5	115	1	0.10302826D-08
5	105	3	0.23134970D-35	5	115	2	0.44745972D-21
5	105	4	0.23269100D-64	5	115	3	0.31951417D-39
5	106	0	0.99999999D+00	5	115	4	0.34460269D-71
5	106	1	0.70754571D-08	5	116	0	0.10000000D+01
5	106	2	0.37695123D-19	5	116	1	0.83139325D-09
5	106	3	0.95235216D-36	5	116	2	0.27318594D-21
5	106	4	0.48363227D-65	5	116	3	0.13119874D-39
5	107	0	0.99999999D+00	5	116	4	0.71381986D-72
5	107	1	0.57137653D-08	5	117	0	0.10000000D+01
5	107	2	0.23047875D-19	5	117	1	0.67084835D-09
5	107	3	0.39192954D-36	5	117	2	0.16676220D-21
5	107	4	0.10048282D-65	5	117	3	0.53860535D-40
5	108	0	0.10000000D+01	5	117	4	0.14781756D-72

TABLE VII

Y_{ij} (a = 5) (CONT'D.)

5	118	0	0.10000000D+01	5	127	4	0.00000000D+00
5	118	1	0.54126567D-09	5	128	0	0.10000000D+01
5	118	2	0.10178244D-21	5	128	1	0.63043210D-10
5	118	3	0.22106237D-40	5	128	2	0.72465485D-24
5	118	4	0.30600829D-73	5	128	3	0.29646478D-44
5	119	0	0.10000000D+01	5	128	4	0.00000000D+00
5	119	1	0.43668214D-09	5	129	0	0.10000000D+01
5	119	2	0.62113389D-22	5	129	1	0.50828588D-10
5	119	3	0.90711800D-41	5	129	2	0.44164004D-24
5	119	4	0.63330411D-74	5	129	3	0.12140939D-44
5	120	0	0.10000000D+01	5	129	4	0.00000000D+00
5	120	1	0.35228139D-09	5	130	0	0.10000000D+01
5	120	2	0.37899695D-22	5	130	1	0.40978086D-10
5	120	3	0.37215097D-41	5	130	2	0.26912440D-24
5	120	4	0.13102844D-74	5	130	3	0.49710930D-45
5	121	0	0.10000000D+01	5	130	4	0.00000000D+00
5	121	1	0.28417365D-09	5	131	0	0.10000000D+01
5	121	2	0.23121999D-22	5	131	1	0.33034642D-10
5	121	3	0.15264498D-41	5	131	2	0.16397812D-24
5	121	4	0.27101608D-75	5	131	3	0.20350412D-45
5	122	0	0.10000000D+01	5	131	4	0.00000000D+00
5	122	1	0.22921776D-09	5	132	0	0.10000000D+01
5	122	2	0.14104419D-22	5	132	1	0.26629452D-10
5	122	3	0.62597267D-42	5	132	2	0.99900518D-25
5	122	4	0.56040614D-76	5	132	3	0.83294710D-46
5	123	0	0.10000000D+01	5	132	4	0.00000000D+00
5	123	1	0.18487727D-09	5	133	0	0.10000000D+01
5	123	2	0.86025301D-23	5	133	1	0.21464952D-10
5	123	3	0.25664880D-42	5	133	2	0.60855430D-25
5	123	4	0.11584866D-76	5	133	3	0.34086758D-46
5	124	0	0.10000000D+01	5	133	4	0.00000000D+00
5	124	1	0.14910427D-09	5	134	0	0.10000000D+01
5	124	2	0.52461331D-23	5	134	1	0.17301074D-10
5	124	3	0.10520480D-42	5	134	2	0.37066489D-25
5	124	4	0.23942057D-77	5	134	3	0.13946948D-46
5	125	0	0.10000000D+01	5	134	4	0.00000000D+00
5	125	1	0.12024538D-09	5	135	0	0.10000000D+01
5	125	2	0.31988617D-23	5	135	1	0.13944149D-10
5	125	3	0.43116720D-43	5	135	2	0.22574328D-25
5	125	4	0.00000000D+00	5	135	3	0.57055698D-47
5	126	0	0.10000000D+01	5	135	4	0.00000000D+00
5	126	1	0.96965875D-10	5	136	0	0.10000000D+01
5	126	2	0.19502737D-23	5	136	1	0.11237951D-10
5	126	3	0.17667339D-43	5	136	2	0.13746755D-25
5	126	4	0.00000000D+00	5	136	3	0.23337067D-47
5	127	0	0.10000000D+01	5	136	4	0.00000000D+00
5	127	1	0.78188356D-10	5	137	0	0.10000000D+01
5	127	2	0.11888869D-23	5	137	1	0.90564668D-11
5	127	3	0.72379098D-44	5	137	2	0.83702464D-26

TABLE VII

Y_{ij} (a = 5) (CONT'D.)

5	137	3	0.95438157D-48
5	137	4	0.00000000D+00
5	138	0	0.10000000D+01
5	138	1	0.72980579D-11
5	138	2	0.50960029D-26
5	138	3	0.39023602D-48
5	138	4	0.00000000D+00
5	139	0	0.10000000D+01
5	139	1	0.58807539D-11
5	139	2	0.31022383D-26
5	139	3	0.15953767D-48
5	139	4	0.00000000D+00
5	140	0	0.10000000D+01
5	140	1	0.47384492D-11
5	140	2	0.18883190D-26
5	140	3	0.65212477D-49
5	140	4	0.00000000D+00
5	141	0	0.10000000D+01
5	141	1	0.38178362D-11
5	141	2	0.11492934D-26
5	141	3	0.26652056D-49
5	141	4	0.00000000D+00
5	142	0	0.10000000D+01
5	142	1	0.30759304D-11
5	142	2	0.69942713D-27
5	142	3	0.10890912D-49
5	142	4	0.00000000D+00
5	143	0	0.10000000D+01
5	143	1	0.24780735D-11
5	143	2	0.42560885D-27
5	143	3	0.44497155D-50
5	143	4	0.00000000D+00
5	144	0	0.10000000D+01
5	144	1	0.19963222D-11
5	144	2	0.25896200D-27
5	144	3	0.18177561D-50
5	144	4	0.00000000D+00
5	145	0	0.10000000D+01
5	145	1	0.16081484D-11
5	145	2	0.15755031D-27
5	145	3	0.74246376D-51
5	145	4	0.00000000D+00
5	146	0	0.10000000D+01
5	146	1	0.12953913D-11
5	146	2	0.95843106D-28
5	146	3	0.30321597D-51
5	146	4	0.00000000D+00
5	147	0	0.10000000D+01
5	147	1	0.10434111D-11
5	147	2	0.58299048D-28
5	147	3	0.12381319D-51
5	147	4	0.00000000D+00
5	148	0	0.10000000D+01
5	148	1	0.84040727D-12
5	148	2	0.35458599D-28
5	148	3	0.50549936D-52
5	148	4	0.00000000D+00
5	149	0	0.10000000D+01
5	149	1	0.67686856D-12
5	149	2	0.21564617D-28
5	149	3	0.20635453D-52
5	149	4	0.00000000D+00
5	150	0	0.10000000D+01
5	150	1	0.54512904D-12
5	150	2	0.13113619D-28
5	150	3	0.84226340D-53
5	150	4	0.00000000D+00

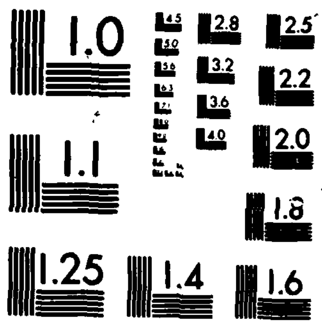
AD-A164 496 MULTICHANNEL COLLISION RESOLUTION ALGORITHM(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA J H CHO DEC 85 2/2

UNCLASSIFIED F/G 17/2 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

LIST OF REFERENCES

1. Tannenbaum, A. S., *Computer Networks*, Prentice-Hall, 1977.

2. Blanc, R. P. and Cotton, I. W., *Computer Networking*, IEEE Press, 1976.

3. Schwartz, M., *Computer-Communication Network Design and Analysis*, Prentice-Hall, 1977.

4. Meditch, J. S. and Chu, W. W., *Computer-Communication Networks*, Pergamon Press, 1983.

5. Rosner, R. D., *Distributed Telecommunications Networks via Satellites Packet Switching*, Wadsworth, 1982.

6. Capetanakis, J. I., "*Tree Algorithms for Packet Broadcast Channels*", *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 505-515, Sept. 1979.

7. Hayes, J. F., "*An Adaptive Technique for Local Distribution*", *IEEE Trans. Commn.*, vol. COM-26, pp. 1178-1186, Aug. 1978.

8. Massey, J. L., *Collision Resolution Algorithms and Random Access Communications*, in Multi-User Commnications (CISM Courses and Lectures No, 265). New York : Springer-Verlag, 1981, pp. 73-137.

9. Georgiadis, L. and Papantoni-Kazakos P., "*A Collision Resolution Protocol for Random Access Channels with Energy Detectors*", *IEEE Trans. Commn.*, vol. COM-30, No.11, Nov. 1982.

10. Gallager, R. G., "*Conflict Resolution in Random Access Broadcast Networks*", in Proc. *AFOSR Workshop Commn.* Theory Appl., Provincetown, MA, Sept, 17-20, 1978, pp. 74-76.

11. Tsybakov, B. S. and Mikhailov, V. A., "*Slotted Multiaccess Packet-Broadcasting Feedback Channel*", *Problemy Peredachi Inform.*, vol.14, No.4, Oct.-Dec. 1978.

12. Mosely, J., *An Efficient Contention Resolution Algorithm for Multiple Access Channels*, Lab. Inform. Decision Syst., M.I.T., Cambridge, MA, Rep.LIDS-TH-918, May 1979.

13. Kleinrock, L., *Queueing System, Volume I: Theory*, John Wiley & Sons, Inc., pp. 103-107, 1975.

14. Abramson, N., "*The ALOHA System - Another Alternative for Computer Communications*", *AFIPS-FJCC*, vol. 37,pp. 281-285, Nov. 1970.

15. Roberts, L. G., *ALOHA Packet System with and without Slots and Capture*, ASS Note 8. ARPA Network Infor. Crr., Stanford Res. Inst., Stanford, CA, June 1972.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5100	2
2.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
3.	Department Chairman, Code 62 Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5100	1
4.	Prof. Jin-Fu Chang, Chairman Dept. of Electrical Engineering National Taiwan University Taipei, Taiwan 107 Republic of China	2
5.	Prof. Hung-Mou Lee, Code 62 Lh Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5100	1
6.	Maj. Joo-Hyung Cho PO #17 Daebang-dong Dongjakgu Seoul 151-01 Republic of Korea	7
7.	LT. Ham, Byung Woon 602-00 Chinhae Naval Academy Faculty E.E Republic of Korea	1

END

FILMED

3-86

DTIC